

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, Journal of GIScience & Remote Sensing, 44(4): 320-333.

**Title:** WMS Performance and Client Design Principles

**Authors:**

<sup>1, 3, \*</sup> Phil Yang, <sup>1</sup>Ying Cao, <sup>2, 3</sup>John Evans

**Affiliations:**

<sup>1</sup>Joint Center for Intelligent Spatial Computing, College of Science, George Mason University, 4400 University Dr., Fairfax, VA, 22030-4444 USA

<sup>2</sup>Global Science & Technology Inc., 7855 Walker Dr., Suite 200, Greenbelt, MD 20770 USA

<sup>3</sup>NASA Geoscience Interoperability Office, Goddard Space Flight Center, Code 604.0 Greenbelt, Maryland, 20771 USA

**Corresponding author:**

Phil Yang, [cyang3@gmu.edu](mailto:cyang3@gmu.edu), Phone: 703-993-4742 or 301-286-5329, Fax: 703-993-9299

**Acknowledgements**

This paper is supported through a NASA *Geosciences Interoperability* grant to George Mason University and an FGDC project on *Building Interoperable Web Mapping Portal for the Middle Atlantic Region*. Comments provided by Professor John R. Jensen, the Editor in Chief of GIScience & Remote Sensing, and the anonymous reviewers are gratefully acknowledged.

**Abstract:** The OGC Web Map Service facilitates sharing geospatial information via the World Wide Web. However, this approach presents performance challenges because some Web Mapping servers are not well optimized for real-world demands, and some client applications are not well designed for performance. To address this problem, we 1) test the performance of different WMS servers; 2) analyze the test to find performance patterns; 3) utilize the patterns to design principles; 4) develop a prototype to test the principles.

**Key Words:** WMS, Interoperability, Performance, Web Services, Spatial Web Portal

## **1. Introduction**

Web-based geospatial interoperability rests on specifications such as the OGC Web Map Service (WMS, de La Beaujardière, 2004). This has seen widespread implementation; for example, NASA's Jet Propulsion Laboratory (JPL) has built a WMS server to share a variety of geophysical data (JPL, 2006). The US Geological Survey (USGS) National Map uses WMS to serve satellite and aerial imagery online (USGS, 2006). Private firms such as Intergraph, ESRI, CubeWerx, and many others, as well as open-source software projects, include WMS software components in their product line, and have developed and maintained WMS services (Intergraph, 2006). WMS is also used as a portrayal service for other OGC web services (Compusult, 2007), such as the Web Feature Service (WFS, Vretanos, 2002) or Web Coverage Service (WCS, Evans, 2003). The hundreds of WMS services deployed have driven the deployment of portals to integrate different WMS services in a data and service sharing environment.

However, when WMS is utilized to support portals and operational systems, performance surface up as an issues. This paper examines patterns in WMS performance, and proposes ways to improve client design. Section 2 describes the general architecture of WMS applications. Section 3 reports WMS performance results under stress testing and mentions methods for WMS server performance improving. Section 4 presents principles for improving WMS client performance. Section 5 presents an applet-based prototype illustrating promising performance changes. Conclusions and discussion follow in Section 6.

## **2. WMS Application Architecture**

The WMS interface consists of three operations: GetCapabilities, GetMap, and GetFeatureInfo. Server responses are either XML documents (for GetCapabilities and GetFeatureInfo) or images (for GetMap). GetCapabilities requests a server's data layers, legend, projections, and other service metadata. GetFeatureInfo returns attributes associated with a given point on a map. GetMap requests a map for a specific geographic location, with a given set of layers and rendering options. To assess WMS performance, it is worthwhile to examine the entire process of an application using WMS – which may include a number of WMSs shared through different servers in the sequence diagram illustrated in Fig. 1.

Fig. 1 illustrates a typical WMS operation sequence:

- 1) a client requests GetCapabilities from one or more WMS servers;
- 2) the server(s) return capabilities responses;
- 3) the client interprets and processes these responses;
- 4) the client initializes its user interface (e.g., to display a legend);
- 5) the client sends GetMap request(s);
- 6) the server(s) return GetMap responses; and finally
- 7) the client processes the images received to create a composite map.

Users can then perform interactive manipulations such as Zoom In, Zoom Out, Pan, and Query. On each of these, the client sends a new GetMap request to each server, receives the image reply, and processes it. Such a procedure presents several potential bottlenecks:

- overall user perceived response time,
- processing responses on the client side,
- network bandwidth, and
- server side processing.

Optimizing system performance requires addressing these bottlenecks systematically. In this paper, we deal with server- and client-side processing bottlenecks by examining the performance patterns of WMS servers, using them to design a high-performance client, and build a prototype to test the performance improvement. Network bandwidth is not researched in this paper because it is concerned with network infrastructure.

### **3. WMS Performance Patterns**

A network GIS performance is the level of Quality of Services, which is mainly defined by the user-perceived response time. The performance of a WMS server is defined by the response times of its three operations: GetCapabilities, GetMap, and GetFeatureInfo, especially when a large number of users are accessing the server simultaneously. Our analysis focuses on the first two operations (the third one is less accessed), GetCapabilities and GetMap. First, we compare the responses from a number of servers to a single request. Next, we “stress-test” several different servers; and finally we analyze performance patterns and sketch performance improving principles.

We selected six popular WMS servers used in many interoperability testbeds, such as the series of OGC Web Services (OWS), and conducted our tests against the six WMS servers listed in Table 1. An initial test against the servers found that 5, 10, 30, 50, and 100 simultaneous requests would reveal useful information. To reduce the impact of client side processing, we recorded only the time spent between sending a request and receiving the response. The tests were conducted with a client running on a computer with a 100Mbps connection, 2 GHz CPU, and 1GB RAM.

#### **3.1 GetCapabilities Performance**

Figures 2a to 2f illustrate the six servers' response times to 1, 5, 10, 30, 50, and 100 simultaneous GetCapabilities requests.

The servers responded to a single request with delays ranging from 0.4 seconds to 5 seconds (Fig. 2a). They responded to five simultaneous requests with delays shown in Fig. 2b. With 10 simultaneous requests, all servers showed a slower response, with some servers taking over 10 seconds (Fig. 2c). With 30 simultaneous requests, some servers took up to 350 seconds before returning error messages (Fig. 2d). With 50 simultaneous requests, some servers had response times that exceeded the client's timeout value. With 100 simultaneous requests, only one server returned valid responses.

The test showed that some servers respond in a timely fashion and are scheduled very well; but that increasing the number of simultaneous requests can greatly increase response times. When the number of simultaneous requests exceeds a certain threshold, some servers blocked all requests. The results suggest that client side applications might improve the user experience by increasing requests to servers with better performance and decreasing the number of requests issued to servers with longer response time. The results also suggest that some servers need optimization to serve a large number of users. Research on WebGIS performance and technologies (such as Yang et al., 2005) addresses this need. Also, the GetCapabilities response can be also cached or buffered to improve the performance so a server don't have to read capability files each time.

To accommodate these performance patterns, WMS clients can: 1) issue simultaneous GetCapabilities requests to all servers, 2) update the user interface after each response, 3) set a timeout limit for responses, and 4) display a progress indicator when waiting for response.

### **3.2 GetMap Performance**

We tested GetMap performance using 1, 5, 10, 30, 50, and 100 simultaneous requests against all servers except one server (which is not stable in responding to GetMap request).

For a single request, server response times ranged from 2 seconds to 30 seconds (Fig. 3a). For 5 simultaneous requests, servers' response times are all increased (Fig. 3b). 10 simultaneous requests resulted in longer response times from all servers, with some responses exceeding the timeout period (Fig. 3c). 30, 50, and 100 simultaneous requests further increased server response times, but some much more slowly than others (Fig. 3d, 3e, 3f); a few servers stopped responding at all.

The test found, first, that the GetMap operation is more time consuming than GetCapabilities. GetMap involves more server side processing, such as selecting data, visualizing each data layer, and resizing & cutting the generated image as map. The test also showed that some servers are very stable with many simultaneous requests, and that some servers may perform very well but stop responding after a certain number.

Many performance improving techniques, such as pyramid-cut-hash, progressive transmission, and clustering (Wang, 2004; Yang et al., 2005), can be adopted to improve the GetMap performance. The patterns revealed from the test are also helpful in designing a client with good performance.

## **4. Client Design Principles**

These performance patterns suggest some principles for the design of WMS applications and clients, so as to provide an acceptable interaction performance. A WMS client usually accesses several servers to populate the legend for selecting layers (from GetCapabilities responses), initialize a map view (from GetMap responses), and manipulate maps through further WMS requests and responses. In this process, timely

display of the initial interface is preferable, which requires a fast WMS server. Therefore, it's good to *draw on fast WMS servers for client initialization*.

Because not all servers are fast, clients are suggested to *show a progress bar when waiting for a slow reply*.

A client that issues requests and processes replies sequentially can become very slow when accessing multiple WMS servers or layers. Simultaneous access to servers and layers can prevent this; therefore clients should *use multithreading to access different layers and servers in parallel, and to display each reply as soon as it is received*.

Second, when requesting several layers from a single server to be overlaid consecutively, clients should *combine multi-layer requests from the same server as one request, and 3) responses should be processed and displayed once each one is received*.

Integrating these principles (summarized in Table 2) into a client results in a procedure as follows (Fig. 4):

- 1) Send GetCapabilities to all servers simultaneously,
- 2) Using multithreading techniques, update the user interface any time a response is received and processed
- 3) Send multi-layer GetMap requests to each server simultaneously whenever feasible.
- 4) Process and display GetMap responses as soon as each one is received, and
- 5) Update the GUI map once any GetMap response was rendered.

Other map operations, such as Zoom and Pan, also use GetMap, and can benefit from these principles as well.

## **5. A Client Prototype**

To test and confirm these principles, we developed a prototype to explore global geophysical sciences and detailed urban maps using the servers tested (Table 1). The

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, *Journal of GIScience & Remote Sensing*, 44(4): 320-333.

performance was tested and compared between using the principles and without using the principles. The client begins by fetching server GetCapabilities; and follows up with GetMap requests. The coordination strategy composes GetCapabilities and GetMap requests and issues requests to the servers through the communication component (Fig. 5).

The application interface follows the principles proposed in section 4 and includes map operation menus on the top, a legend on the left side, a progress bar on the bottom, and the map area (Fig. 6a). We tested its performance on a computer with 100 Mbps connection to the Internet, 2 Ghz CPU, and 1G RAM.

Once the application was started, the interface's legend was updated after 187 ms, 625 ms, 750 ms, 4703 ms, and 4922 ms when all principles were taken into account (Fig. 6b). The initial map was returned within 547 ms after initializing the legend. This process produced 7 updates in 6 seconds. If layers are not displayed as soon as they return from servers, there's only one update at the 5<sup>th</sup> second and one update around 5.5 second when all responses are received. And if the client makes sequential rather than parallel requests, there will be only one update at approximately the 15<sup>th</sup> second, for a much longer response time.

When selecting one layer from each service to customize a map (Fig. 7a), the response times range from 344 ms to 8.4 seconds with 6 updates when started. It may take up to 8 seconds to get one update if layers are not displayed as soon as they return from servers. The same responses will take up to 17 seconds with one update if requests are sequential rather than parallel.

The two examples show that the user interaction experience can be greatly improved by adopting the above principles to the design of a WMS client.

An interoperable geospatial system (such as the Geospatial One-Stop Portal, FGDC, 2004, or the Earth Science Gateway, NASA GIO, 2004) includes four parts (Fig.

8): 1) legacy components, such as ESRI services and MAP06, developed for a specific application, 2) interoperable web services, such as WFS and WCS, added to share the existing components, 3) facilitating services, such as WCPS and WMS overlay, and 4) a client with which end users may interact with the portal's information resources. The performance improving principles adopted in the WMS client can also be used to support the facilitating services (Fig. 8) as detailed in Table 3. Besides this, using a fast WMS for initial display within WMS overlay and Displaying a progress indicator in the user interface (Viewer) could also help with user interaction.

## **6. Conclusion and Discussion**

This paper reports tests performed against several WMS servers to reveal performance patterns. It proposes a set of principles for the design of a high-performance WMS client. It describes results from a prototype illustrating the use of these principles. The principles can also be used in other interoperable geospatial information systems, such as spatial web portals, to improve performance.

More generally, these principles are relevant to the performance of all systems sharing geospatial information. For example, each of the OGC Web Services specifications needs to be researched to improve the performance, for example, using binary coding to transmit GML in WFS (Vretanos, 2002). In a complex geospatial information sharing environment such as GEOSS (GEO, 2005) or National Applications identified by NASA (Birk, 2005), performance affects the entire procedure from data collection (earth observation), data analysis (earth system modeling/simulation), and application logic (decision supporting tools), to end users, especially when real time decision making is needed, such as real time routing (Cao, 2007).

Another way to improve performance is to apply emerging computing technologies, such as Grid computing, to parallel processing massive geospatial processing (Aydin et al., 2005) to facilitate complex application.

## References

1. Aydin G., Aktas M. S., Fox G. C., Gadgil H., Pierce M., Sayar A., 2005, "SERVOGrid Complexity Computational Environments (CCE) Integrated Performance Analysis," In *Proceedings of Grid Computing Conference*. Seattle, Washington, USA, 13-14 November 2005. 256 – 261.
2. Birk R. J., 2005, Extending NASA Earth-Sun System research results to serve GEOSS societal benefits, In *Proceeding of the 3rd International Workshop on the Analysis of Multitemporal Remote Sensing Images*, May 2005.
3. Cao Y., 2007, *Near Real-Time Transportation Routing Supported by Grid Computing*, Dissertation, George Mason University. 104 p.
4. Compusult, 2007, *Web Enterprise Suite (WES): Developer's Guide*, Mount Pearl, Newfoundland, CA. 119 p.
5. de La Beaujardiere J. eds, 2004, *OGC Web Map Service Implementation Specification 1.3*, [[http://portal.opengis.org/files/?artifact\\_id=5316](http://portal.opengis.org/files/?artifact_id=5316)].
6. Evans J. D. eds, 2003, *OGC Web Coverage Service Implementation Specification 1.0*, [[http://portal.opengeospatial.org/files/?artifact\\_id=3837&version=2](http://portal.opengeospatial.org/files/?artifact_id=3837&version=2)].
7. FGDC, 2004, *Geospatial One-Stop*, [<http://geodata.gov/>].
8. GEO, 2005, *GEOSS 10 year plan*, [<http://www.grss-ieee.org/managedfiles/PDF/GEOSSImplementation.pdf>].

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, *Journal of GIScience & Remote Sensing*, 44(4): 320-333.

9. Intergraph, 2006, *Intergraph WMS*, [<http://www2.demis.nl/>].
10. JPL, 2006, *NASA JPL WMS*, [<http://wms.jpl.nasa.gov/>].
11. NASA GIO, 2004, *Earth Science Gateway*, [<http://esg.gsfc.nasa.gov/>].
12. USGS, 2006, *The National Map*, [<http://nationalmap.gov/>].
13. Vretanos P. A. eds, 2002, *OGC Web Feature Service Implementation specification 1.0*, [[http://portal.opengeospatial.org/files/?artifact\\_id=7176](http://portal.opengeospatial.org/files/?artifact_id=7176)].
14. Wang P., Yang C., Yu Z., Ren Y., 2004, *A Load Balance Algorithm for WMS*, In *IGARSS Proceeding*. pp. 2920-2922.
15. Yang C. and Evans J., 2006, Network GIS Performance, in the *Encyclopedia of GIS* (H. Xiong and S. Shekhar eds.), Springer, (in press).
16. Yang C., Wong D., Yang R.X., Li Q., Tao V., Kafatos M., 2005. Performance-improving techniques in web-based GIS, *International Journal of Geographic Information Sciences*, 3(19): 319-341.

## Tables

**Table 1 WMS Services Selected for the Research and Their Characteristics**

Server Label	Service Link	Characteristics	Usage in the Application
A	<a href="http://nmcatalog.usgs.gov/catalogwms/base">http://nmcatalog.usgs.gov/catalogwms/base</a>	Detailed information about populated locations within US	Provided detailed maps for US
B	<a href="http://www2.demis.nl/mapserver/request.asp">http://www2.demis.nl/mapserver/request.asp</a>	Limited data services with very good performance	Serve as introductory service to shorten user's response time
C	<a href="http://wms.jpl.nasa.gov/wms.cgi">http://wms.jpl.nasa.gov/wms.cgi</a>	A variety of scientific geophysical image datasets	Provide geoscience data services
D	<a href="http://globe.digitalearth.gov/viz-bin/wmt.cgi">http://globe.digitalearth.gov/viz-bin/wmt.cgi</a>	Good global view data, such as LandSat TM data	Provide good global view datasets
E	<a href="http://132.156.10.87/cgi-bin/atlaswms_en">http://132.156.10.87/cgi-bin/atlaswms_en</a>	Detailed map of Canada	Provide detailed map for Canada
F	<a href="http://152.61.128.152/servlet/com.esri.wms.Esrimap/world">http://152.61.128.152/servlet/com.esri.wms.Esrimap/world</a>	Popular service with limited accessibility	Demonstrate the integration of popular services

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, Journal of GIScience & Remote Sensing, 44(4): 320-333.

**Table 2 How to Integrate the Principles into WMS Client Design**

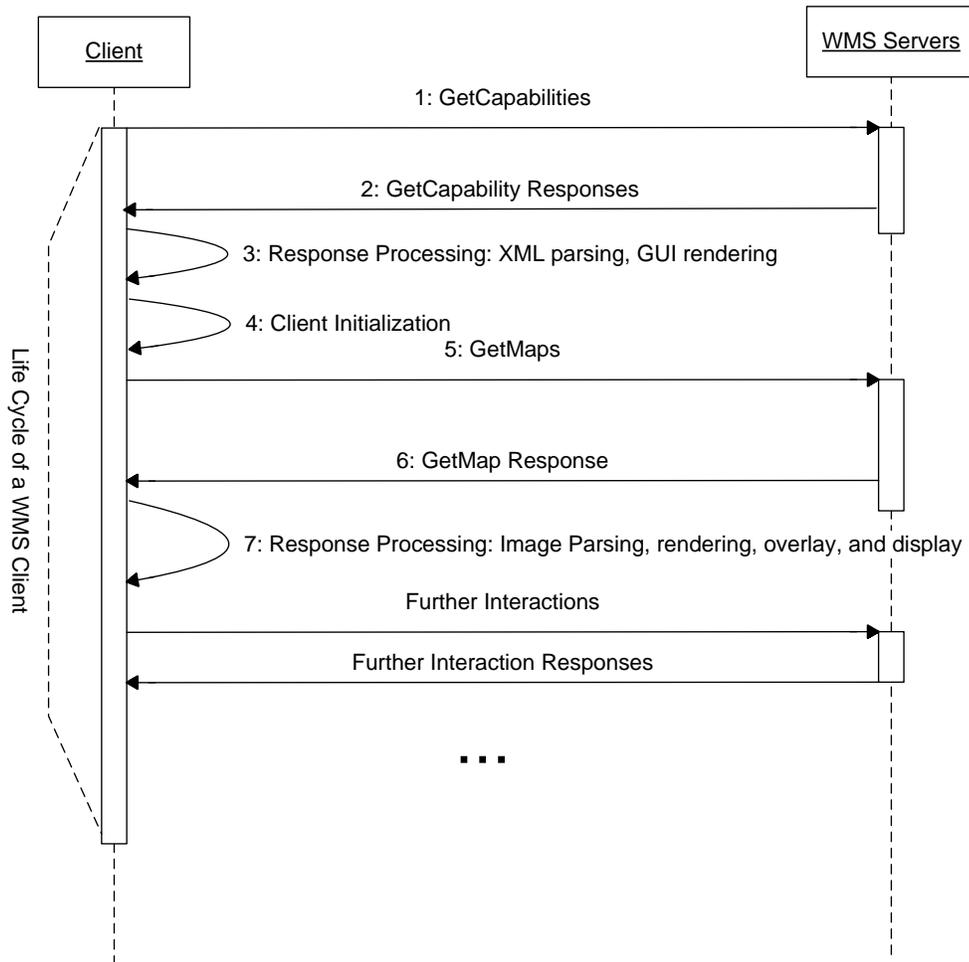
<b>Principles</b>	<b>Design Integration</b>
1	Issue simultaneous GetCapabilities and GetMap requests to all servers
2	For a server with more than one layer to be displayed sequentially, issue a single multi-layer request
3	Process responses and update the user interface as soon as each response is received

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, Journal of GIScience & Remote Sensing, 44(4): 320-333.

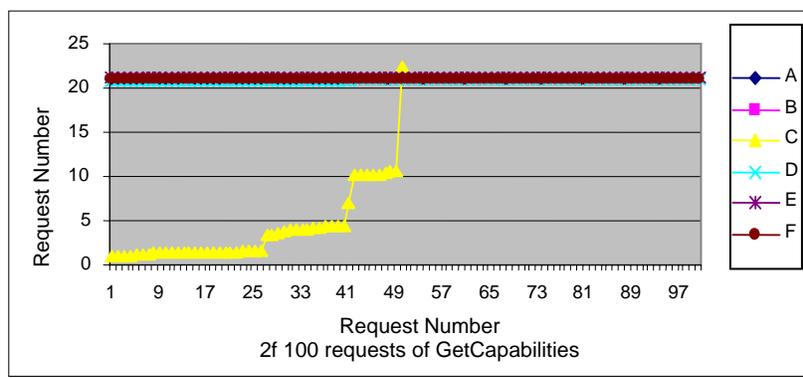
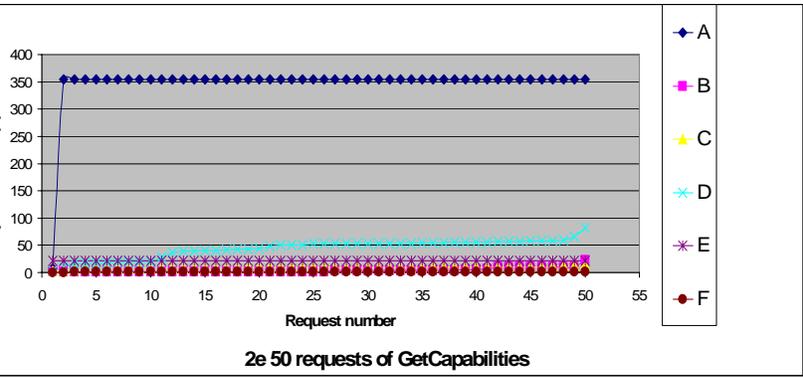
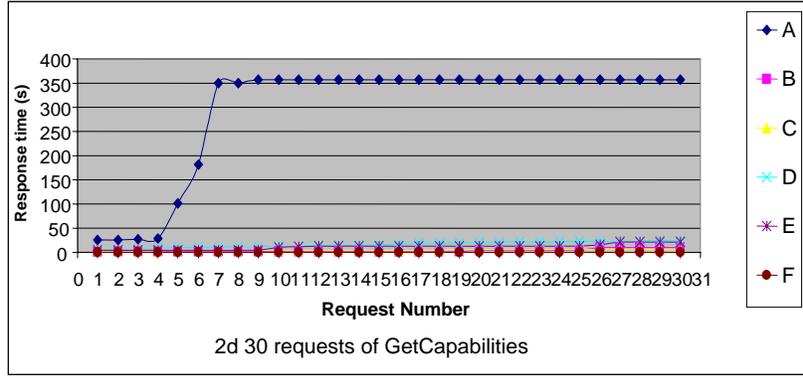
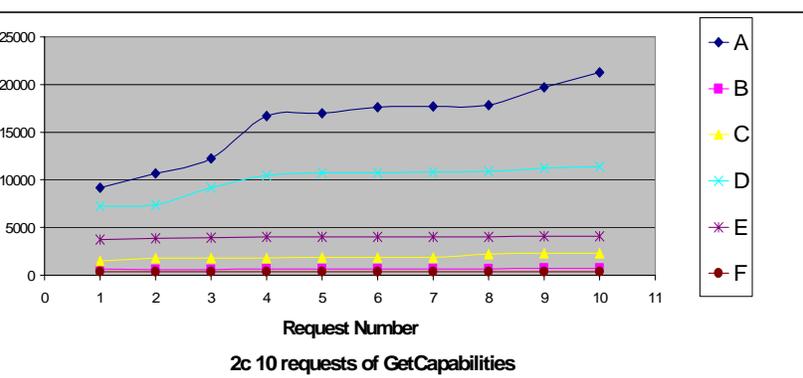
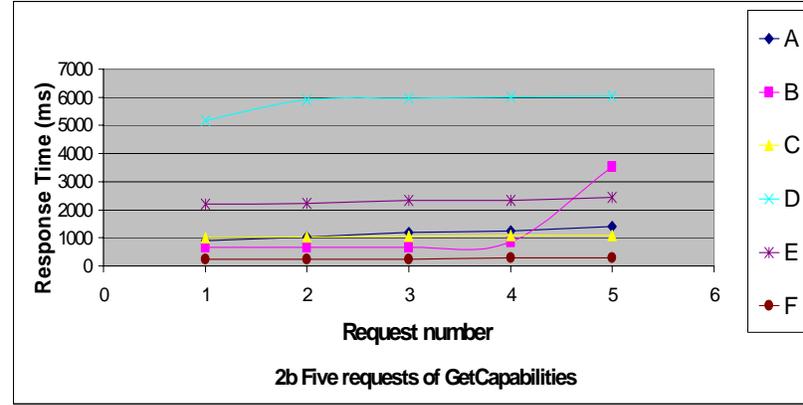
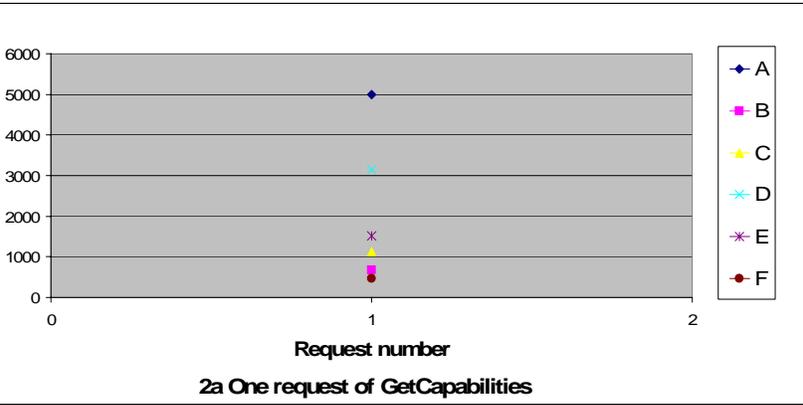
**Table 3 How to Apply Principles in Different Components for Facilitating Services**

<b>Principles</b>	<b>Applicable Components</b>	<b>Usage</b>
1	WMS Overlay, WMS Bridge, WCPS, WFPS	Issue simultaneous GetCapabilities and GetMap requests to all servers
2	WMS Overlay, WMS Bridge, WCPS, WFPS	For a server with more than one layer to be displayed sequentially, issue a single, multi-layer request
3	WMS Overlay and the Viewer	Process responses and update the user interface as soon as each response is received

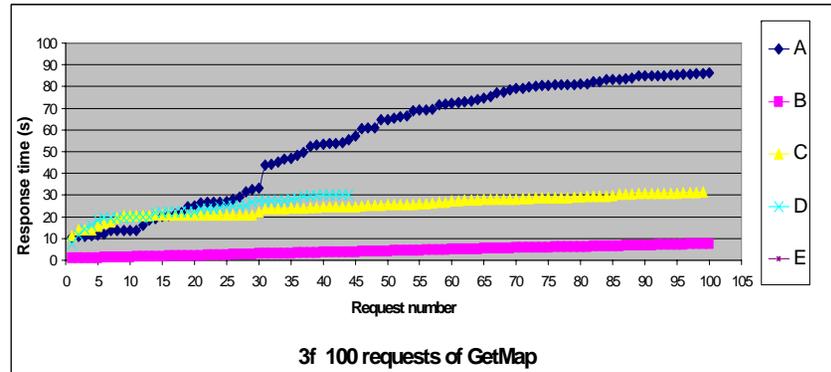
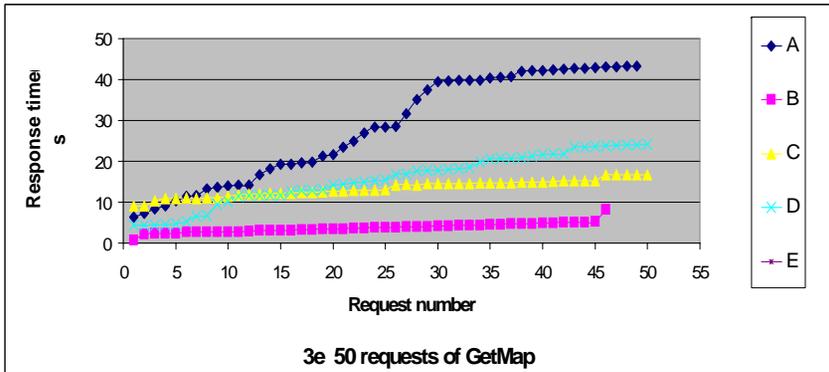
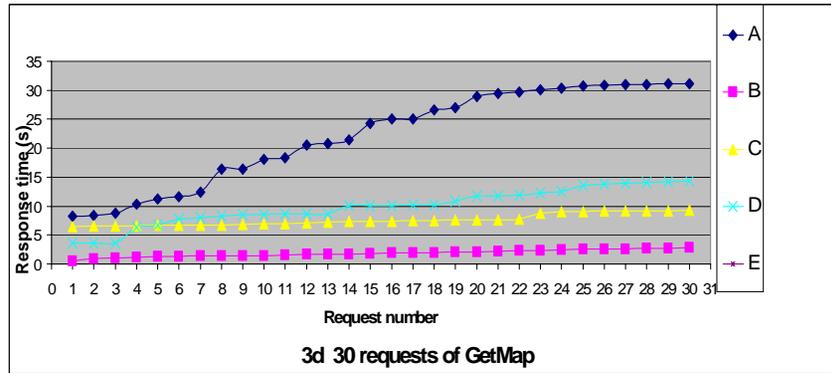
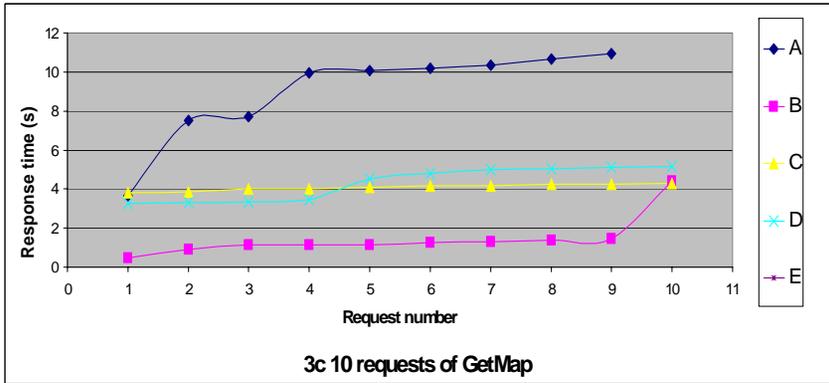
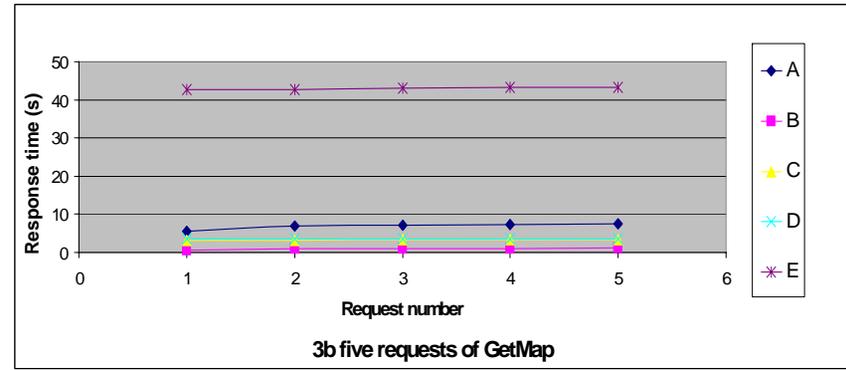
## Figures



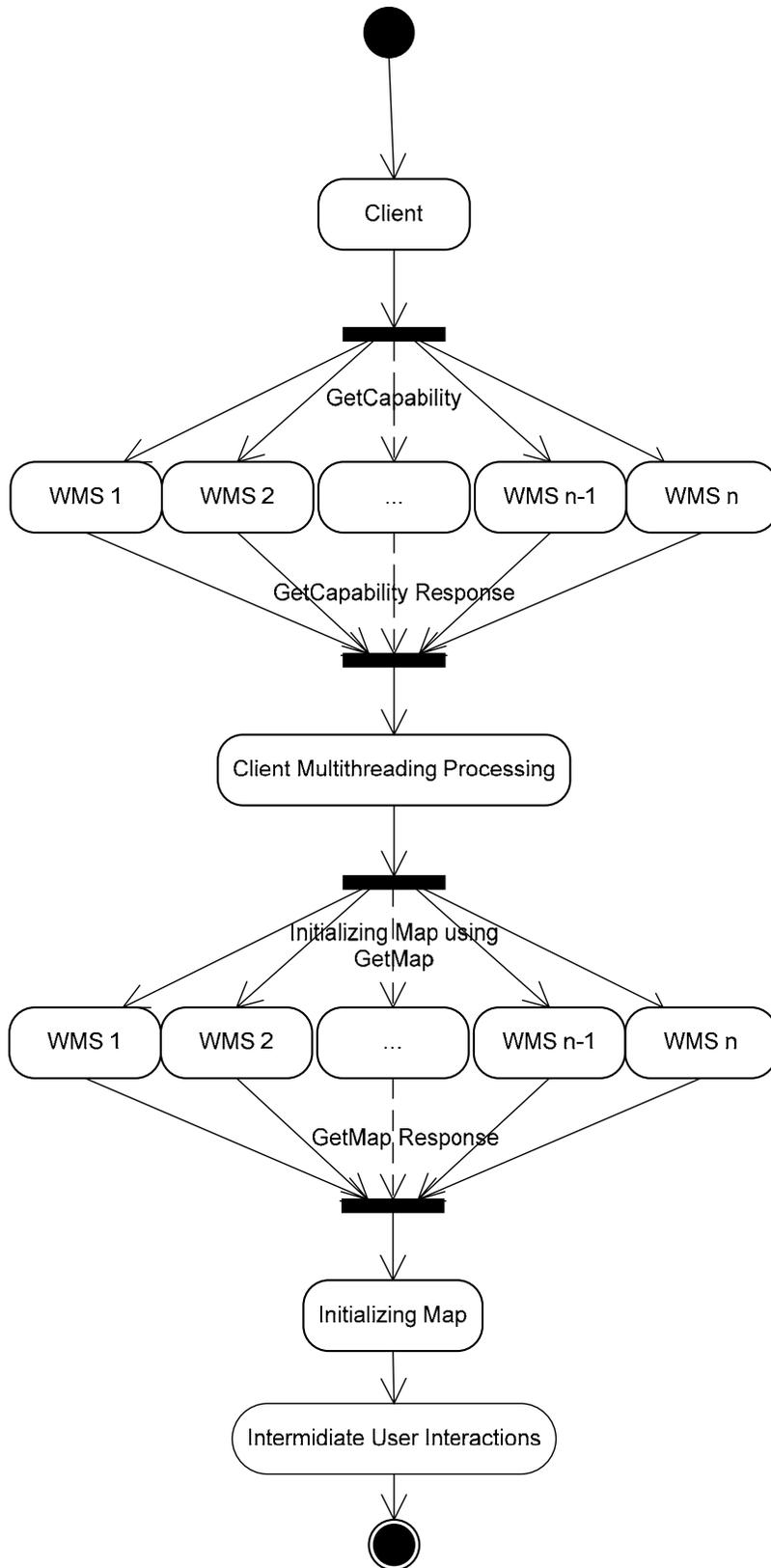
**Fig. 1.** The WMS application sequence.



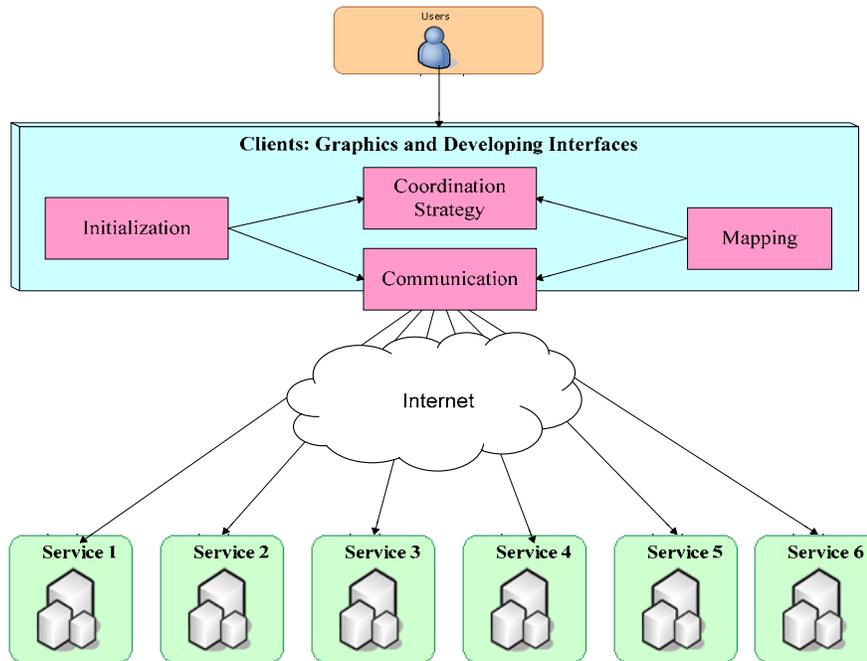
2. The response times of different WMS servers when issuing different number of simultaneous GetCapabilities requests (x axis is the responding sequence and y axis is the response time in seconds). A validation of the response contents shows that the flat pattern of National Map in 2d, 2e, and flat pattern of all in 2f except F are not valid responses).



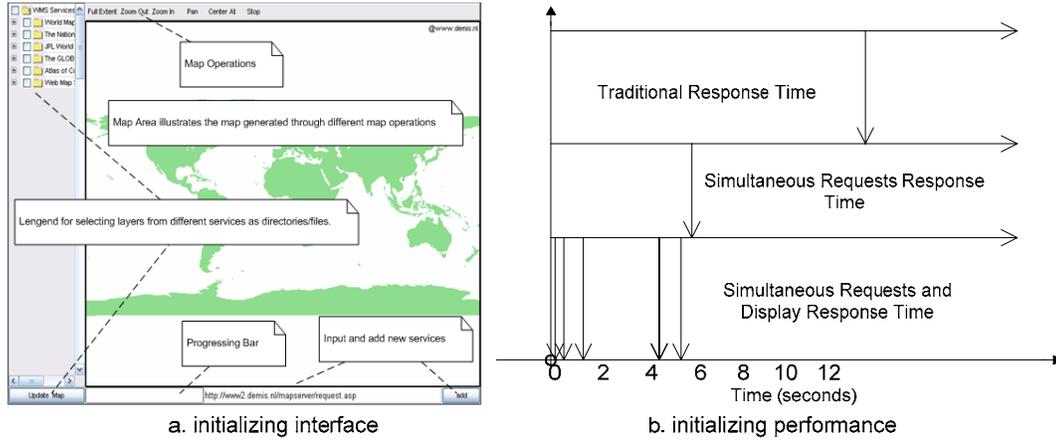
**Fig. 3.** The response times of the six WMS servers when issuing different number of simultaneous GetMap requests (The x axis is the returning sequence; the y axis is the response time in seconds).



**Fig. 4.** A flow chart of WMS application integrating the principles in Table 2.

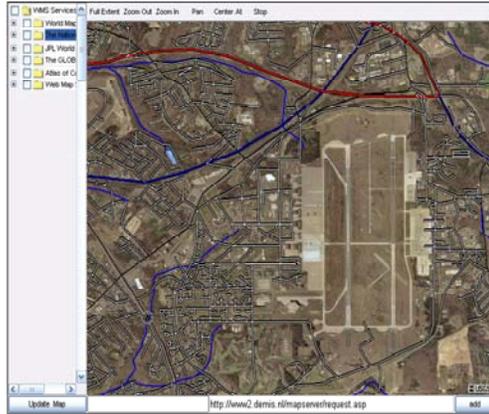


**Fig. 5.** The architecture of the prototype.

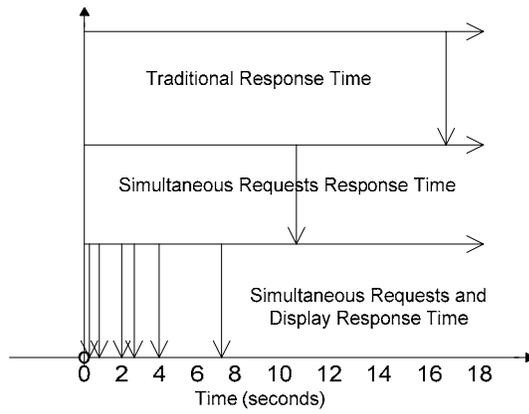


**Fig. 6.** The initialized interface and response time when different principles are taken.

Yang, Cao, and Evans, 2007, WMS Performance and Client design principles, Journal of GIScience & Remote Sensing, 44(4): 320-333.

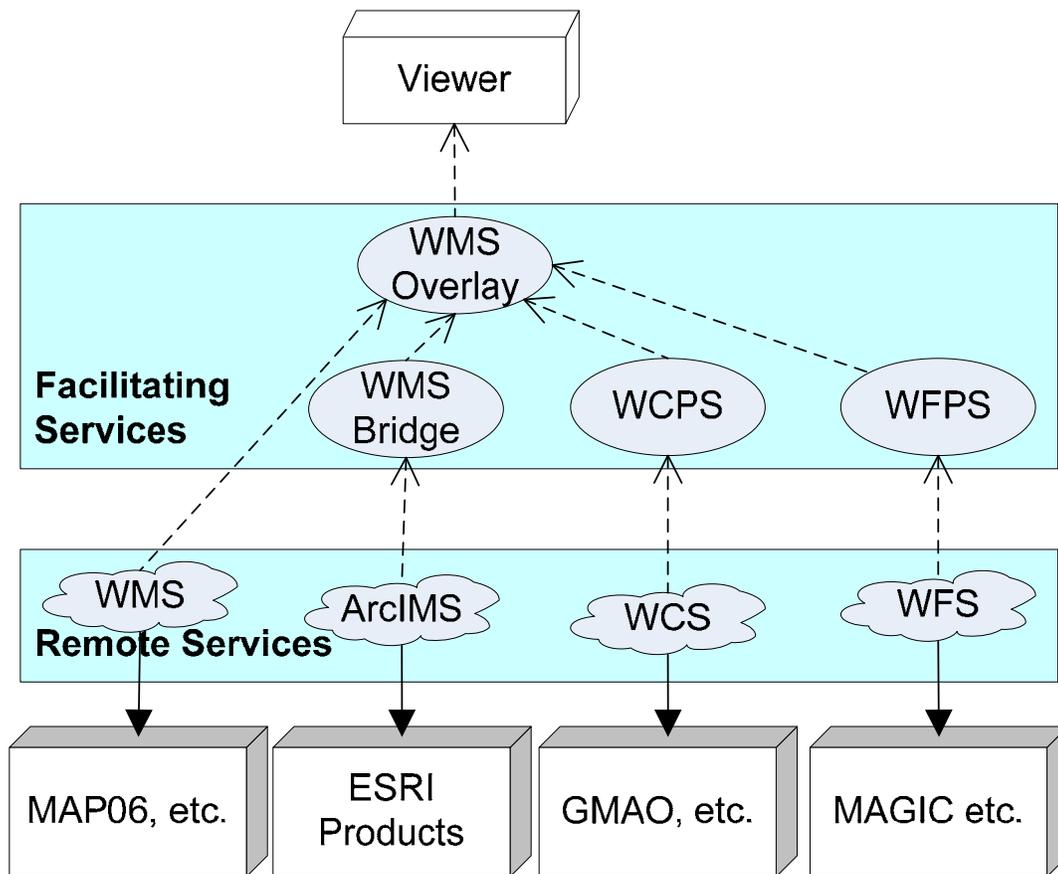


a. mapping interface



b. mapping performance

**Fig. 7.** The customized map and response times when different principles are taken.



**Fig. 8.** WMS within an interoperable architecture.

WCPS: Web Coverage Portrayal Service

WFPS: Web Feature Portrayal Service

ArcIMS: an ESRI product supporting Web Mapping

MAP06: Modeling, Analysis, and Prediction Program of 2006

GMAO: Global Modeling and Assimilation Office

MAGIC: Middle-Atlantic Geospatial Information Consortium