Impact of User Concurrency in Commonly Used Open Geospatial Consortium Map Server Implementations

Joan Masó, Paula Díaz, Xavier Pons, José L. Monteagudo-Pereira, Joan Serra-Sagristà, Francesc Aulí-Llinàs
Universitat Autònoma de Barcelona, Spain
joan.maso@uab.cat, paula.diaz@uab.cat, xavier.pons@uab.cat, jlino@deic.uab.cat, joan.serra@uab.cat, fauli@deic.uab.cat

***Abstract*—In emergency situations, it is of paramount importance that accurate assessments showing what is happening in the field, and when and where it is happening, are distributed quickly. This increases the aid workers' awareness of the situation and can be used to organize the workers more efficiently. The increasing number of satellite images available means that new data can be obtained rapidly and the information can be kept constantly up to date. These data can be distributed easily using open standards over the Internet. In a large post-disaster event, the demand for information increases dramatically, which can negatively impact the performance of the services provided. Here, we assess seven of the most popular server solutions (GeoServer, MapServer, MiraMon Map Server, Express Server, ArcGIS Server, TileCache and GeoWebCache) for map service standards (WMS, WMTS, WMTS-C, TMS), and compare their response times, user functionalities and usability.**

***Keywords-server; WMS; tile; response time; cluster; performance*.**

I. INTRODUCTION

Nowadays, there is an increasingly large amount of data, software and geographic standards available (public, private and voluntary), which allow satellite data to be used in a wider range of consolidated and specialized areas and applications. Current space technologies, such as meteorological and earth observation satellites integrated in global networks like GMES, communication satellites and Global Navigation Satellite Systems (GNSS) combined with Geographical Information Systems (GIS) [1], hazard modeling and analysis have also contributed to this increase in applications and data. Nevertheless, better spatial, temporal (synergies, constellations, etc.) and spectral resolutions of remote sensing imagery generate a huge amount of data that is difficult to store, discover, analyze and distribute. Heaps of tapes, CDs and hard drives full of data have been replaced by web-based data dissemination infrastructures that make searching and discovery easier. Web portals and clearinghouses increasingly implement standardized protocols and are integrated into a larger System of Systems, like GEOSS [19].

Despite the number of map server implementations that claim to be the fastest and the most robust on the market, there are few studies that apply rigorous metrics to determine the real performance of the servers or compare strategies to increase their performance.

This paper is an extension of a previous article [11] that evaluates the efficiency and possibilities of several map servers (i.e., MapServer [7], GeoServer [4], MiraMon Map Server [18], Express Server [6], ArcGIS Server [3], TileCache [13, 8] and GeoWebCache [15]) that implement international standards (e.g., Web Map Server (WMS [2]), Web Map Server – Cache (WMS-C [13]), Web Map Tile Server (WMTS [9])) connected to satellite image repositories. This research was carried out in the context of GEO-PICTURES, an acronym for *GMES and Earth Observation combined with Position based Image and sensor Communications, Technology for Universal Rescue, Emergency and Surveillance*. GEO-PICTURES is an EU FP7 SPACE project that aims to integrate satellite imagery into in-situ sensors and geo-tagged media (photos and video) to create a tool for decision making in emergency disaster situations. The complete GEO-PICTURES solution covers the capture, transmission, and analysis of data, which is re-elaborated and re-distributed to the aid forces as well as to the general public using several web platforms.

This article begins with the description of the materials and methodology used to perform this study, followed by a through explanation of the tests applied to the performance of the servers. Here an evaluation of concurrent requests to a single server and to a cluster of servers is done. The article continues with the comparison of different standards, this is what it is called: tiling the request and response. Finally, it concludes with a section where the most relevant results are discussed.

II. MATERIALS AND METHODOLOGY

Trials were performed with 22 GeoEye-1 (Orthorectified GeoTIFF; provided by Google [5]) imagery datasets that form a 4Gb raster of 40994 x 57392 pixels, covering Port-au-Prince and surroundings, on 16 January, 2010, 3 days after the earthquake. The influence of scale, intensive client use, and image size and format (JPG, GIF and PNG) were studied for the WMS, WMS-C and TMS [16, 17] protocols. As possible solutions to concurrent requests, we evaluated the efficiency of the Internet cache as well as a cluster of servers in an NLB (Network Load Balance) configuration. The seven servers were set with the minimum configuration required to be run, i.e., without any extra preparation of the data. In order to shield probabilistic error caused by network latency and other uncertain factors, the products were considered to be deployed and requested by local clients [14].

In order to guarantee the comparability of the results, the seven server software (Table 1) were installed on the same computer, which acted as a common server (Intel® Core™ i3 CPU 540 @ 3.07 GHz, 3.06 GHz, 2.92 GB de RAM. Microsoft Windows XP Professional, version 2002 Service Pack 3). Requests were randomly generated from clients in the local network.

TABLE I. LIST OF THE SERVERS EVALUATED AND THEIR SPECIFICATIONS.

Server	**Specifications**	**Date**
MapServer	version 5.6.3 over Apache web server version 2.2.15	April 20, 2010
GeoServer	version 1.7.2 over Jetty web server version 6.1.8	January 19, 2008
ArcGIS Server	version 9.3.1 over Internet Information Service version 5.1	January 1, 2009
Express Server	version 6.1 over Internet Information Service version 5.1	July 1, 2008
MiraMon Map Server	version v. 7.0e over Internet Information Service version 5.1	July 26, 2010
TileCache	version 2.11 over Internet Information Service version 5.1	December 22, 2007
GeoWebCache	version 2.0.1 over Jetty web, 'build over GeoServer	January 20, 2010

MapServer, GeoServer and ArcGIS Server can work over the original GeoTIFF images without having to create an image mosaic. MapServer and GeoServer require a shape file to be created with rectangles representing bounding boxes of each raster file (index file), and ArcGIS Server requires Image Definition (ISDef), which makes it possible to use the original GeoTIFF image format provided by GeoEye-1. Express Server requires the image index to be in JPEG2000 or in the MsSID compressed format. MiraMon requires a full automated pre-rendering of the images in several resolutions to set up the images. This takes several minutes to complete and needs up to 33% extra disk space. TileCache and GeoWebCache can create pre-rendered tiles and save them in a cache for further use or generate them on the fly. Automatic on the fly generation is an advantage that can save time when a new layer is set up in an emergency situation.

All studied protocols request maps by creating an URL using specific standardized syntax. This URL is requested from the server and an image is obtained in screen resolution (in the case of error the server sends an exception message). The URL requests were randomly generated by a program and requested using a command line tool (an application called *Wininet*) that waits for the response, saves it and reports the total time spent on this particular communication. The response time was redirected to an archive that was converted into a table of values that were used to create the performance graphics that are shown here. This paper describes the methodology employed and the numerical and graphical performance metrics, and evaluates strategies for improving performance.

The randomly generated URL methodology employed guarantees that speed measures are comparable and independent from the selected bounding box or request sequence. Nevertheless, users in front of a computer screen browsing the maps do not generate random requests but rather they request regions next to the previous ones at the same zoom level (pan) or they zoom in and out in the same region. However, human browsing patterns are out of the scope of this work and will be considered in the future.

III. EVALUATION OF CONCURRENT REQUESTS TO A SINGLE SERVER

One of the main factors that affect the performance of web servers is the concurrency of requests. We measured both the influence of the pixel size and the image size on the response time for WMS requests. More than one hundred different requests were made from up to 6 simultaneous clients. The graph (Figure 1) shows the response time for different pixel size requests.

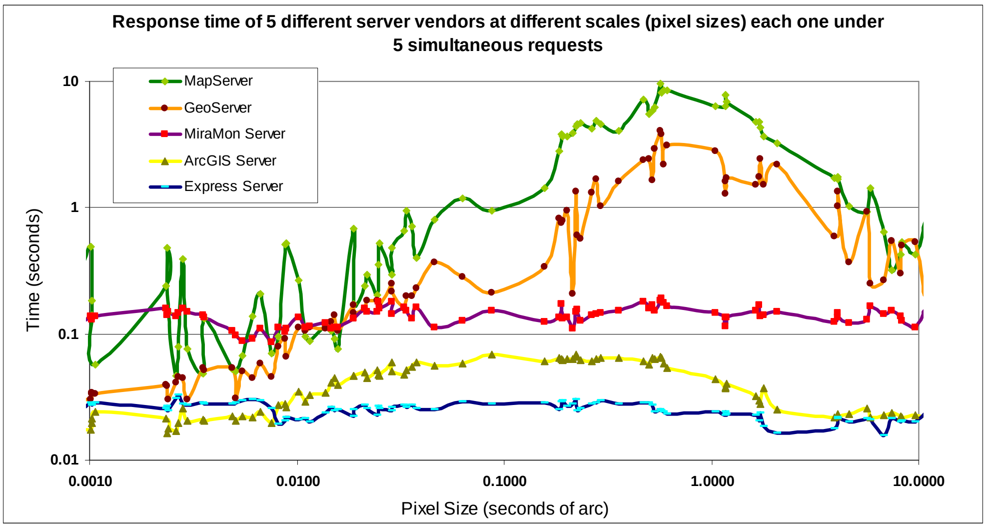

Figure 1. The response times of MapServer, GeoServer, MiraMon Map Server, ArcGIS Server and Express Server in relation to pixel size in concurrent requests from 5 simultaneous clients respectively.

One common aspect of the servers analyzed is that the response time increases as the number of simultaneous requests made by clients to a single server increases. The fastest server is Express Server, probably due to the nature of the wavelet compressed format (MrSID or JPEG2000) that is internally organized in a pyramid of zoom levels of the images used as a database. ArcGIS Server obtains the best results using original GeoTIFF images. MiraMon Map Server obtains intermediate results as it requires a pre-rendering process to generate tiles. MapServer is programmed in C language and GeoServer uses java code. MapServer performs faster when a single client is used, but GeoServer is faster than MapServer for concurrent requests. This could be because java provides better and easier multithread support. There are also small differences in response times depending on the output format requested. JPEG is the fastest format in MapServer, Express Server, ArcGIS Server and MiraMon Map Sever, but PNG is fastest in GeoServer.

A request with a pixel size that generates a map covering a region equivalent to the boundary of the GeoTIFF set (nearly 0.893 seconds of arc in range, width 443) obtains the slowest response time. A map with a smaller pixel size only shows a part of the GeoTIFF set area and obtains a faster response. A map with a larger pixel size leaves some blank areas, and also results in a faster response time.

IV. EVALUATION OF A CLUSTER OF SERVERS

In the current state of maturity of the hardware it is not possible to dramatically increase performance by getting a faster machine, even if you are willing to pay more. Current computer technology has reached a speed limit in CPU processing time and disk speed access. To overcome the performance degradation observed in concurrent requests a possible solution is to set up a cluster of servers that can act as a virtual single server that deals with requests in parallel. We carried out tests comparing a single WMS server (Figure 2) and a WMS computer cluster server (Figure 3), in which 6 computers are able to respond to different clients at the same time as if they were a faster single server. These tests (consisting in the same requests) were carried out with up to 17 simultaneous clients to evaluate the response time of the MiraMon Map Server.

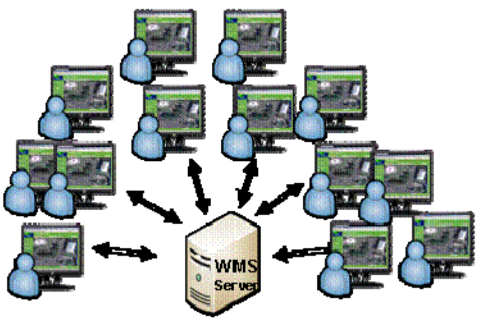

Figure 2. Computer single server structure.

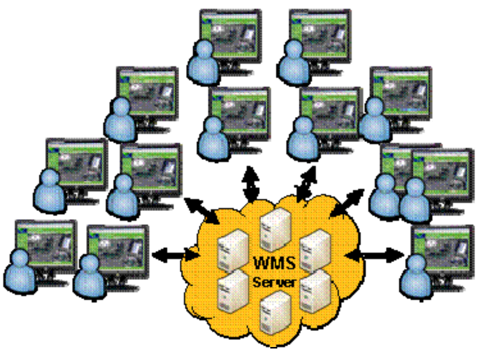

Figure 3. Computer cluster server structure.

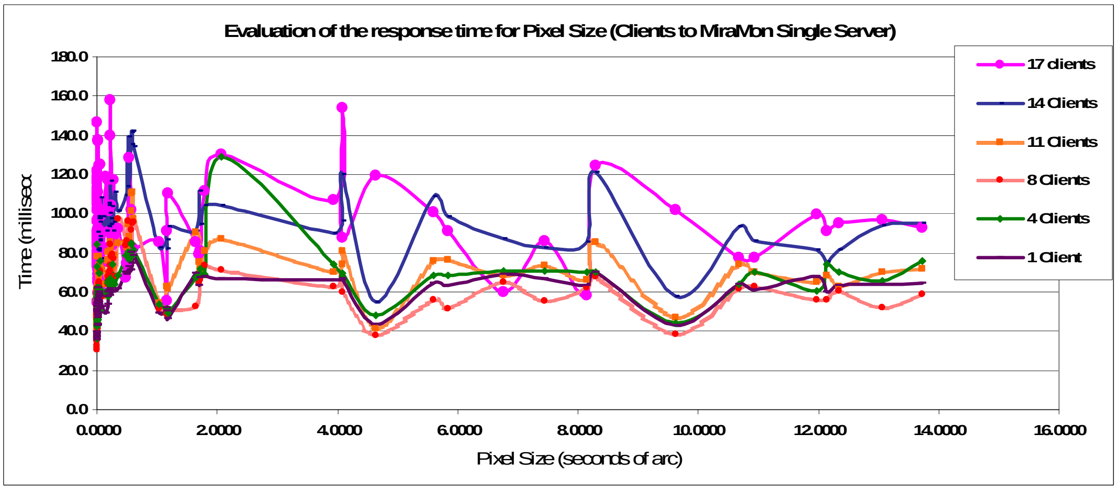

Figure 4. Response time for different concurrent requests for up to 17 clients of a single MiraMon Map Server.

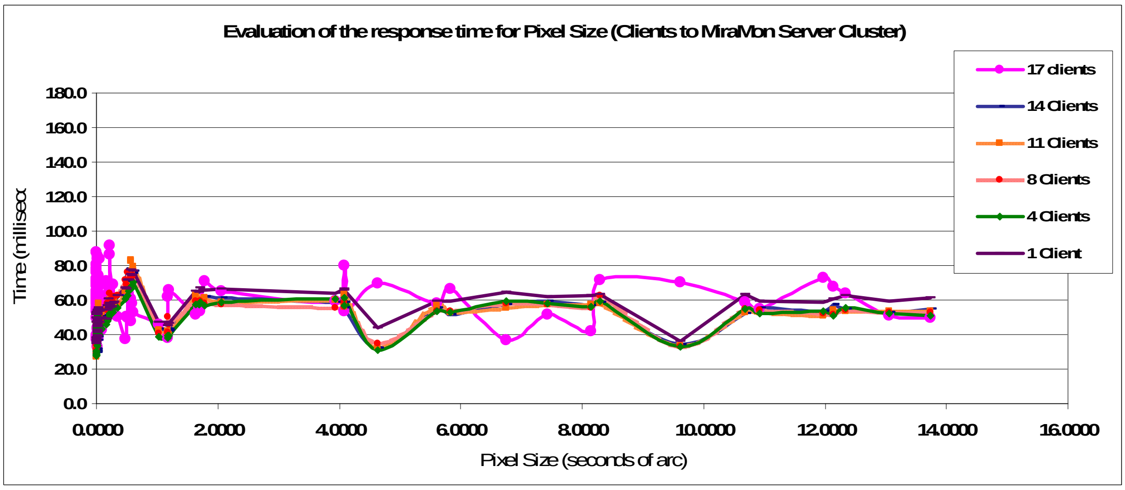

Figure 5. Response time for different concurrent requests for up to 17 clients of a cluster MiraMon Map Server.

Response time measurements comparing a single server (Figure 4) and a six-computer cluster (Figure 5) stressed with multiple client requests show that the response time of the NLB server is much more stable and almost equivalent to the single client stress case, even for 17 simultaneous requests. We expect some degradation if we increase the number of requests further, but fortunately the performance of the NLB cluster can be improved again by aggregating more servers to the cluster (up to 64 in Windows 2003). If we suppose that the performance is linear, this means that this configuration can be scaled to serve at least ~200 simultaneous requests without performance degradation. Note that the response time for these requests is always lower that 0.1 second, so this configuration is equivalent to 2000 requests per second.

V. TILING THE REQUEST AND RESPONSE

In the previous sections, we assumed a common WMS interaction in which a WMS client requests the entire image needed to cover the client viewport in a single piece. Some WMS clients (like OpenLayers) are now able to tile the space in a regular matrix of small pieces [12]. Therefore, several tiles are needed to cover the whole viewport but the client can recycle some tiles when the user moves the view laterally and can also take advantage of the cache mechanisms. However, this strategy can have its drawbacks if the caching mechanism cannot help and the server has not been prepared to manage this situation because, as we have discussed previously, the response time can increase even if each tile is smaller than the whole view. However, users do not perceive this because some tiles get to the client sooner and are shown immediately. This paper clarifies the effects of this approach on a classical WMS server and quantifies the difference between fast full image delivery (WMS) and tiled image delivery (WMS-C and TMS). We also studied improving these situations by applying tile strategies directly to the server, like OSGeo WMS-C and OGC WMTS (10).

We carried out speed metrics in 3 different services for 7 servers: Express Server, ArcGIS Server, MiraMon Map Server, GeoServer, MapServer, TileCache combined with MapServer and GeoWebCache combined with GeoServer. We simulated tiled clients (tiles of 256x256 pixels) that make requests to common WMS (which have no particular strategy for dealing with tiles) in the three following configurations:

- Tiled WMS in unlimited concurrent requests. This consists in requesting all the tiles needed to cover the viewport at the same time. In our case, from 6 to 12 tiles were needed to cover the entire viewport requested. In some cases, this resulted in momentary server saturation (Figure 6), like in MapServer and GeoServer. The three servers with the best performance were Express Server, ArcGIS Server and GeoWebCache.

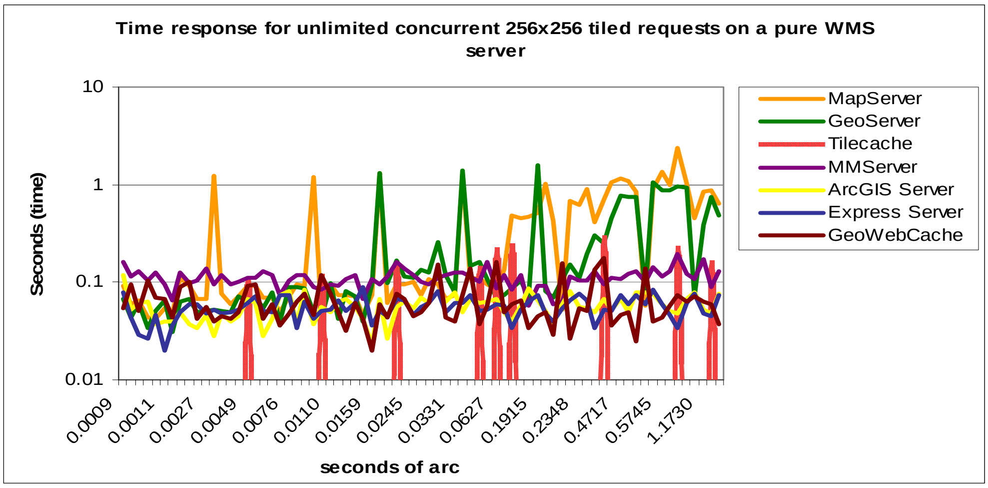

Figure 6 . WMS-C for unlimited concurrent tile requests.

- Tiled WMS in semi-concurrent requests. This consists in limiting simultaneous requests to the maximum number of requests to a server that a web browser allows (e.g., Firefox 3.6 allows 6 simultaneous petitions but Internet Explorer 6.0 only allows 2). In our case, we used a mean value of four tiles at the same time, then we waited until the server finished to request the next four (Figure 7). Some servers performed better compared to the previous case, such as MapServer, GeoServer and MiraMon, while others performed worse. Tiled servers performed better in general.

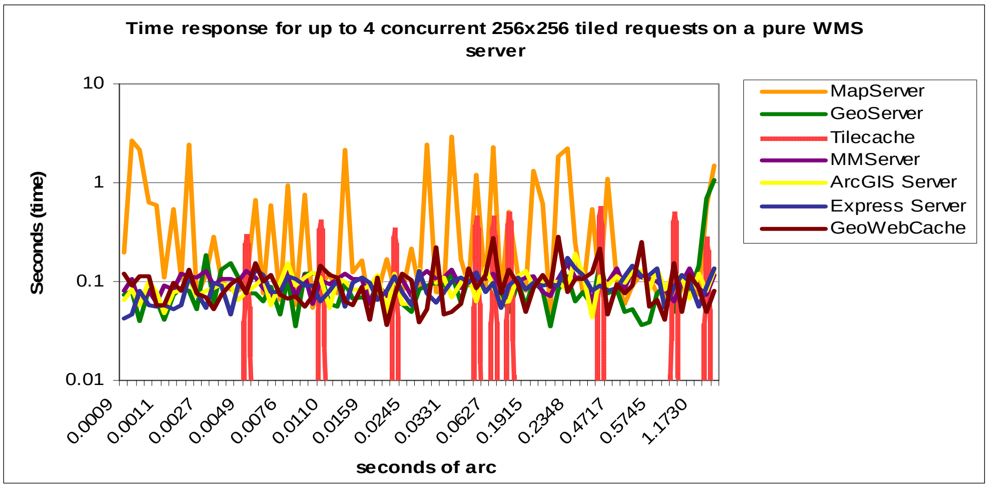

Figure 7.WMS-C for semi-concurrent requests, up to 4 concurrent tile requests.

- Tiled WMS in sequential requests. This consists in requesting each tile after the previous request has been completed (Figure 8). This results in a more stable response time but it is not the optimum situation, especially for GeoServer, MiraMon Server and in some cases ArcGIS Server. GeoWebCache has the best performance.

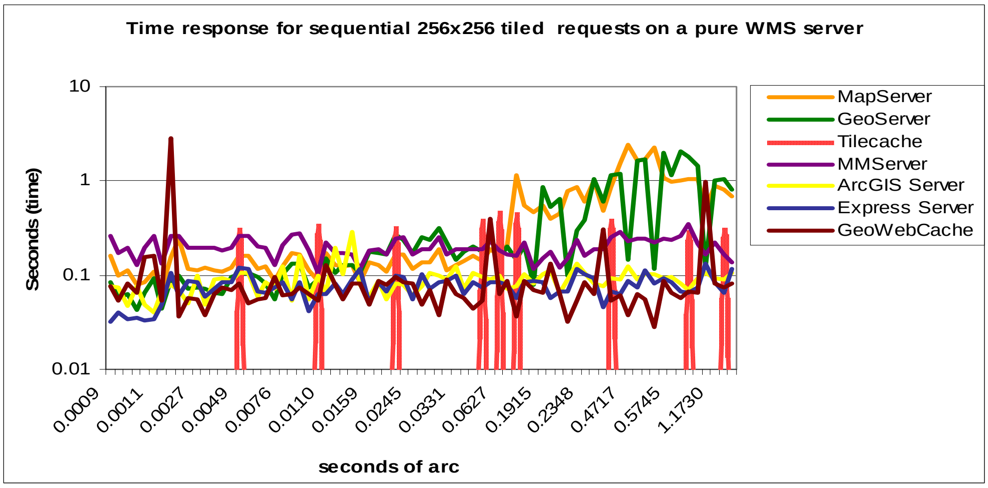

Figure 8. WMS-Cached for sequential requests.

Finally, we compared these three configurations with a regular WMS full viewport image (Figure 9), and evaluated performance degradation. TileCache and GeoWebCache are not represented in Figure 9 because these servers are not able to respond to a full WMS viewport.

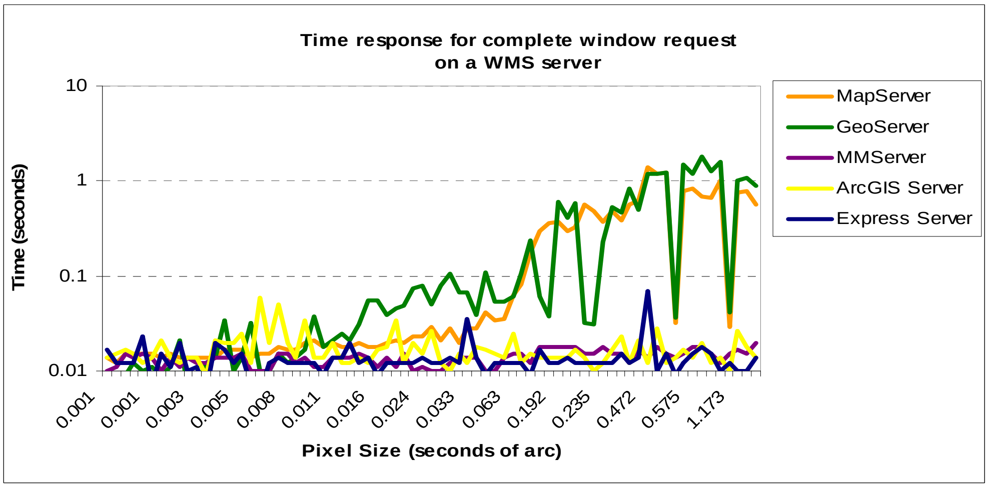

Figure 9. Regular WMS for full image requests.

Figure 9 shows that a full WMS viewport is the fastest for all servers, particularly for Express Server, ArcGIS Server and MiraMon Map Server, probably because the server only does the work once even if the volume of information to deliver is bigger. When tiles are used, requesting all tiles sequentially results in the slowest solution for all servers; however, limiting the number of concurrent requests to 4 improves the response time significantly. This is the best performance situation for MiraMon Map Server and GeoServer. After seeing this, it is easy to understand why many web browsers limit the number of simultaneous requests to a relatively small number (depending on the product and the version). Out of the concurrent tile request situations, this is the best tile solution for MapServer, Express Server and ArcGIS Server. Determining the optimum semi-concurrence number for each server will be the focus of a future work.

The tile products tested (TileCache and GeoWebCache) provide a way of pre-rendering tiles or saving tiles that are generated on the fly for further use. The main drawback is the generation time, but this can be partially overcome by *metatiling* strategies. Both TileCache and GeoWebCache support metatiling. Instead of generating each tile individually, a *metatile* is generated, creating a single large map image that can be divided into a number of tiles. Figure 10 shows that for all servers, analyzing a 512 x 512 image requires more time, but much less time than that required for analyzing four 256 x 256 images. This is because generating a metatile involves accessing source data only once instead of four times for a set of four tiles

 ISBN: 978-1-61208-161-8

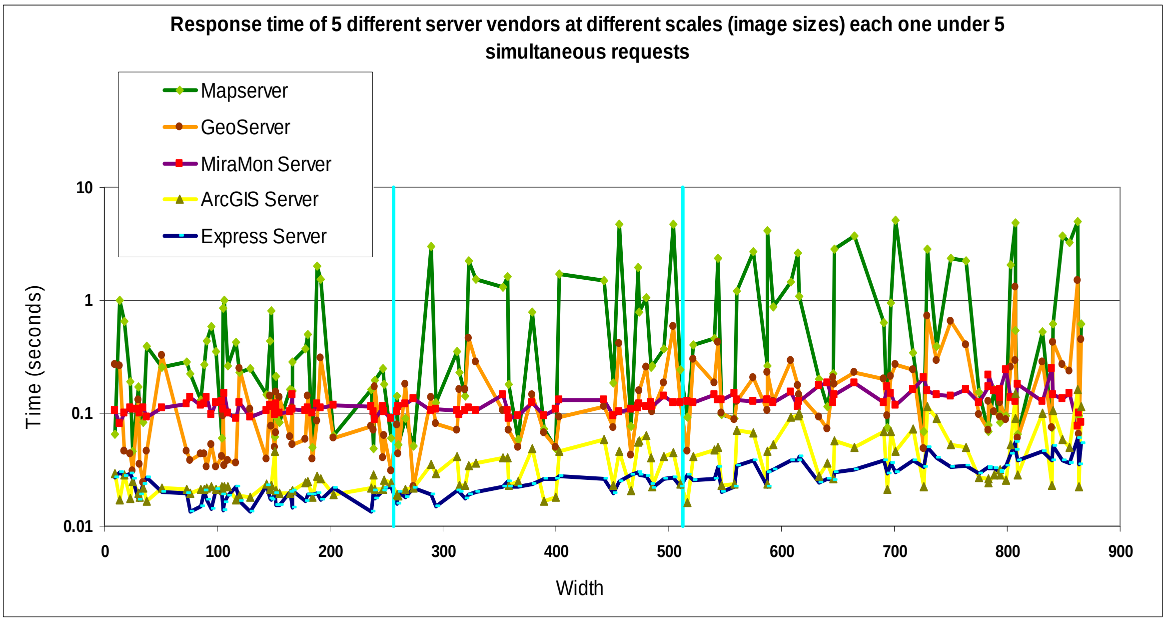

Figure 10. Response time for image size in concurrent requests to 5 single servers. Two marks at 256 and 512 pixel sizes have been added to facilitate the comparison between the response time in these two common pixel sizes.

VI. CONCLUSIONS

The speed tests described in this paper are a practical demonstration of the suitability of certain servers and service configurations in certain domains in which the reliability of the services under high stress conditions is imperative. This document summarizes and quantifies the results of our speed tests and determines which servers are faster under the minimum configuration.

All the analyzed servers have slower performances when the number of simultaneous clients is increased. A cluster server can be used to solve this situation: a group of computers is able to respond at the same time to different clients, assigning each client to a different computer in the group.

The results show that WMS servers do not perform well if clients using tile strategies are used over servers that are not optimized for tile response. MapServer and GeoServer with minimum data configuration do not require a data preparation process; however, they do not perform as well as other services that require indexing methods like MiraMon Map Server or Express Server. MapServer (based on C++ code) performs better than GeoServer (based on Java code) under single client requests, but GeoServer is surprisingly faster under concurrent simultaneous requests. The fastest WMS server is Express Server which works with MsSID or JPEG2000 compressed images that are 5% of the original size. The fastest tile server in the three cases assessed (concurrent, semi-concurrent and sequential requests) is GeoWebCache built over GeoServer.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Government, by FEDER, and by the Catalan Government, under Grants TSI2006-14005-C02-02, TIN2009-14426-C02-02 and GRUMETS 2009 SGR 1511, and also by the European Commission through the FP7-242390-GEO-PICTURES project (FP7-SPACE-2009-1). Xavier Pons is the recipient of an *ICREA Acadèmia* Excellence in Research grant.

REFERENCES

[1] Altan, O., Backhaus R., Boccardo, P., and Zlatanova, S. (2010), Geoinformation for Disaster and Risk Management. Examples and Best Practices, Copenhagen, Denmark. ISBN 978-87-90907-88-4.

[2] De la Beaujardiere, J. (2004) – OGC Web Map Service (WMS) Interface. Ver.1.3.0, OGC 03-109r1. Web: http://www.opengeospatial.org/standards/wms, last access March 1st 2011.

[3] ESRI (2010), Image Server & ArcGIS Server, 9.3. 380 New York St., Redlands, CA 92373-81000 USA.

[4] GeoServer (2010) user manual version 1.7, Web: http://docs.geoserver.org/, last access March 1st 2011.

[5] Google Crisis Response (2010), Haiti Earthquake – GeoEye Satellite Imagery Download, Web: http://www.google.com/relief/haitiearthquake/geoeye.html, last access March 1st 2011.

[6] LizardTech (2010), LizardTech Inc y Lizardtech España SL. GeoExpress compressor. version 8 & Express Server™ version 6.1.

[7] Map Server Team, (2010), MapServer Release 5.6.5 documentation, Open source Web Mapping, Web: http://mapserver.org/tutorial/index.html, last access May 24th 2011.

[8] Marin, L., (2008) Setting up TileCache on IIS, in a blog at WorldPress.com Web: http://viswaug.wordpress.com/2008/02/03/setting-up-tilecache-on-iis/, last access March 1st 2011.

[9] Masó J., Pomakis K., and Julià N. (2010a) – OGC Web Map Tile Service (WMTS). Implementation Standard. Ver 1.0, OGC 07-057r7 Web: http://www.opengeospatial.org/standards/wmts, last access March 1st 2011.

[10] Masó J., Zabala A., and Pons X. (2010b) Combining JPEG2000 Compressed Formats and OGC Standards for Fast and Easy

Dissemination of Large Satellite Data, Italian Journal of Remote Sensing - 2010, 42 (3): 101-114

[11] Masó J, Díaz, P., and Pons, X. (2011) Performance of standardized web map servers for remote sensing imagery. In: Proceedings of "Data Flow: From Space to Earth. Applications and interoperability Conference" (21-23th March 2011, Venice).Corila - Consorzio per la Gestione del Centro di Coordinamento delle Attività di Ricerca Inerenti il Sistema Lagunare di Venezia. pp.83-83 ISBN:9788889405154.

[12] Matt Mills, NASA World Wind Tile Structure. Web: http://www.ceteranet.com/nww-tile-struct.pdf, last access March 1st 2011.

[13] MetaCarta Labs, TileCache Contributors (2006-2010), TileCache Web: http://tilecache.org/, last access May 24th 2011.

[14] Nie Y., Xu, H., and Liu H., (2011), The Design and Implementation of Tile Map Service, in Advanced Materials Research, Vol. 159 (2011) pp 714-719, Trans Tech Publications, Switzerland.

[15] OpenGeo (2011), GeoWebCache User Manual, Web: http://geowebcache.org/docs/current/, last access May 24th 2011.

[16] OSGeo (2011), Tile Map Service Specification, OSGeo Wiki. Web: http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification, last access March 1st 2011.

[17] OSGeo, OpenLayers wiki, Using Custom Tile Sources, Google-like Tile Layer Support, Web: http://trac.osgeo.org/openlayers/wiki/UsingCustomTiles, last access March 1st 2011.

[18] Pons, X., (2004) MiraMon. Geographic Information Systems and Remote Sensing software. v. 4, Center for Ecological Research and Forestry Applications, CREAF, Bellaterra. 286 p. ISBN: 84-931323-4-9.

[19] GEOSS, The Global Earth Observation System of Systems. Web: http://www.earthobservations.org/