

Masó, J., Julià, N. y Pons, X. (2010): El nuevo estándar internacional OGC-WMTS. Oportunidades de aplicación y rendimiento versus OGC-WMS. En: Ojeda, J., Pita, M.F. y Vallejo, I. (Eds.), *Tecnologías de la Información Geográfica: La Información Geográfica al servicio de los ciudadanos*. Secretariado de Publicaciones de la Universidad de Sevilla. Sevilla. Pp. 1.295-1.303. ISBN: 978-84-472-1294-1

## EL NUEVO ESTÁNDAR INTERNACIONAL OGC-WMTS. OPORTUNIDADES DE APLICACIÓN Y RENDIMIENTO VERSUS OGC-WMS

Joan Masó<sup>1</sup>, Núria Julià<sup>1</sup> y Xavier Pons<sup>2,1</sup>

(1) Centre de Recerca Ecològica i Aplicacions Forestals (CREAF), Edificio C, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona). joan.maso@uab.cat, n.julia@creaf.uab.cat

(2) Departamento de Geografía. Edificio B, Universitat Autònoma de Barcelona 08193 Bellaterra (Barcelona) xavier.pons@uab.cat

### RESUMEN

El uso masivo de servicios Web Map Service (WMS) y la sencillez del estándar ha comportado algunos problemas debido a que la mayor parte de implementaciones tratan las peticiones de manera independiente sin aprovechar las respuestas anteriores para despachar las nuevas rápidamente. Algunos fabricantes vieron estas ineficiencias y aparecieron diversas estrategias que discretizan el espacio. En 2007, el Open Geospatial Consortium (OGC) inició un estudio para estandarizar y unificar todos estos productos que se concretó, en 2010, en el nuevo estándar Web Map Tile Service (WMTS).

El OGC-WMTS describe una geometría de malla regular de teselas para un conjunto de escalas conocidas; introduciendo la capacidad de obtener una tesela de manera compatible con el uso de los mecanismos de caché en Internet. Sin embargo, el WMTS podría no ser adecuado para todas las situaciones. Esta comunicación describe las principales características de WMTS y ahonda en las estrategias para soportar con éxito situaciones como el pre-renderizado, los clusters de servidores de balanceo de carga de red y los servicios de mapas frecuentemente actualizados. También se discuten las posibles estrategias de mejora de rendimiento aprovechando los mecanismos de caducidad del HTTP y de notificaciones de actualización de contenido.

Palabras Clave: mapa, servicio, Internet, rendimiento, tesela.

### ABSTRACT

*The massive use of Web Map Service (WMS) and the simplicity of this standard has resulted in some problems due to the fact that most implementations deal with requests independently without taking advantage of previous responses to dispatch the new ones faster. Some manufacturers saw these inefficiencies and various strategies based on a discretized space were applied. In 2007, Open Geospatial Consortium (OGC) started a process to standardize and unify all these products that, in 2010, crystallized in the new Web Map Tile Service (WMTS) standard.*

*OGC-WMTS describes a regular grid of tiles geometry for a known set of scales; introducing the ability to get a tile compatible with Internet caching mechanisms. However, WMTS may not be suitable for all situations. This paper describes the main features of WMTS and emphasizing in strategies to successfully withstand conditions such as pre-rendered, network load balanced services clusters and services with frequently updated maps. We also discuss possible strategies for performance improvement by taking advantage of HTTP expiration mechanisms and content update notifications.*

*Keywords: map, service, Internet, performance, tile.*

## INTRODUCCIÓN Y ANTECEDENTES

El estándar Web Map Service (WMS) aprobado, en su versión 1.0.0, en abril de 2000 por el Open Geospatial Consortium (OGC) define un servicio Web para producir mapas dinámicamente a partir de datos espaciales referenciados geográficamente. Según los estándares OGC e ISO, un mapa es una representación de la información geográfica en forma de imagen digital en un formato apto para ser visualizado en la pantalla de un ordenador. Los mapas no son los datos geográficos en sí, sino imágenes pictóricas en formatos como PNG, GIF o JPEG. Se definen tres operaciones básicas; una que permite conocer los metadatos del servicio, otra que permite obtener un mapa con un ámbito geográfico y dimensiones conocidas, y una última, opcional, que permite obtener información de los objetos particulares mostrados en el mapa (Beaujardiere, 2004; Masó y Julià, 2008).

El WMS ha sido globalmente aceptado y durante sus 10 años de existencia ha habido una importante proliferación de servicios WMS, en parte gracias al impulso de las infraestructuras de datos espaciales (IDE) o a iniciativas como la directiva INSPIRE, pero también a un diseño razonablemente acertado y sencillo. El WMS ha contribuido, indudablemente, a extender el uso público de la información geográfica a través de clientes web de mapas en Internet.

Cuando se diseñó este servicio, el área de los mapas mostrados en pantalla (normalmente mediante un cliente web de mapas) era relativamente pequeño; pero con el aumento de la velocidad de transmisión de los datos de las redes, con el aumento de la resolución de los monitores y el avance general de los recursos informáticos, se ha propiciado el aumento de las dimensiones y de la resolución del mapa en pantalla y, con ello, el tamaño de los mapas a generar por estos servicios.

El principal inconveniente del WMS reside en su sencillez: raramente hay dos peticiones WMS iguales. En un servicio WMS el mapa se obtiene a partir de un envolvente y del ancho y alto de la ventana de la pantalla en píxeles; esto provoca que cada refresco de pantalla (o de un fragmento si se ha producido un pequeño desplazamiento lateral) comporte una nueva petición WMS de toda la ventana, diferente de las anteriores. Además, todos los usuarios del servicio realizan peticiones, a veces muy similares, pero que no resultan idénticas ni en escala ni en ámbito geográfico, con lo que el servidor no puede aprovechar las respuestas anteriores para despachar rápidamente las nuevas peticiones. Con el aumento del número de usuarios, el servidor recibe diversas peticiones concurrentes para mapas similares pudiendo llegarse al colapso del servicio por exceso de trabajo.

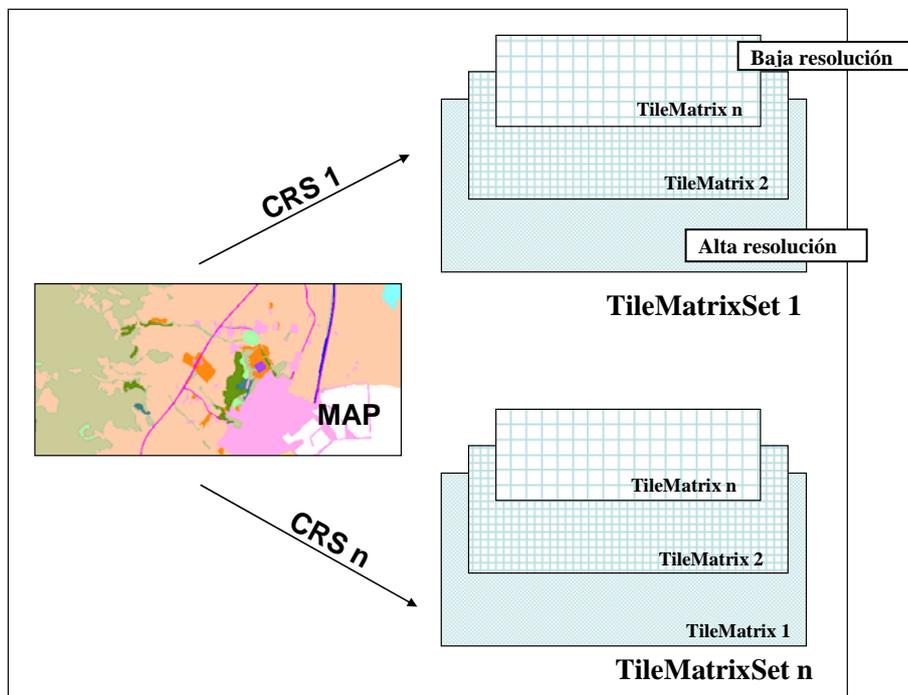
Algunos proveedores se dieron cuenta de que este sistema es ineficiente y no saca partido de los mecanismos de *cache* de Internet (almacenar peticiones idénticas durante un cierto tiempo en servidores intermedios o en el propio cliente), que evitan nuevas consultas al servidor. Aparecieron diversas estrategias que discretizan el espacio geográfico, tanto en tecnologías propietarias (Google Maps o Microsoft Bing/Virtual Earth/Live Maps), como en especificaciones abiertas (WMS-C o TMS de OSGeo). En 2007 el OGC analizó el problema e inició un estudio para determinar la mejor manera de estandarizar todos estos productos bajo un único paradigma. El grupo de trabajo del WMS, en el que los autores de esta comunicación trabajaron activamente, continuaron su desarrollo hasta conseguir el consenso suficiente para editar un nuevo estándar internacional, el cual fue aprobado a principios del 2010 con el nombre del OGC Web Map Tile Service (WMTS) 1.0.0.

## EL NUEVO ESTÁNDAR WEB MAP TILE SERVICE DE OGC

El WMTS de OGC proporciona un enfoque complementario al WMS; mientras el WMS se centra en la renderización de mapas personalizados y es una solución ideal para datos dinámicos, el WMTS renuncia a la personalización de los mapas para obtener una mayor escalabilidad, sirviendo datos prerenderizados donde la envolvente y las escalas han sido restringidas a un conjunto discreto de teselas que siguen una geometría de malla regular. Ortofotos o mapas con poca actualización temporal son ejemplos de capas ideales para elegir WMTS como servicio de difusión de mapas. El conjunto de teselas permite la ejecución de un servicio WMTS utilizando un servidor web que simplemente devuelve los archivos existentes y el uso de mecanismos de red estándar de escalabilidad, tales como sistemas de *cache* distribuidos (Masó *et al.*, 2010a).

Un WMTS es un servicio escalable y *cacheable*, que usa un modelo de teselas (*tiling model*) parametrizado de tal manera que un cliente puede hacer peticiones de un conjunto discreto de valores y recibir rápidamente del servidor fragmentos de imágenes prerenderizadas (*tiles*) que generalmente ya no requieren de ninguna manipulación posterior para ser mostrados en pantalla. Cada una de las capas (*layers*) de un servidor WMTS sigue una o diversas estructuras piramidales de escalas (*tile matrix sets* o conjunto de matrices de teselas), en la que cada escala o nivel de la pirámide (*tile matrix* o matriz de teselas) es una rasterización y fragmentación regular de los datos geográficos

a una escala o tamaño de píxel concreto (ver figura 1) (Masó *et al.*, 2010b). Por ello, una capa puede estar disponible en varios sistemas de coordenadas, y tener diferente ámbito en función de éstos.



**Figura 1.** Modelo piramidal de teselas de una capa de un servidor WMTS.

El estándar define tres recursos básicos: el documento de metadatos del servicio, una tesela (de un determinado *tile matrix set*, a una escala y sistema de referencia determinado) y el documento con la información de un punto sobre una tesela (que contiene información de los objetos particulares mostrados en el mapa), así como tres operaciones básicas para obtener estos recursos: GetCapabilities, GetTile y GetFeatureInfo (opcional) (ver tabla 1) (Masó *et al.*, 2010a; Masó y Julià; 2008).

WMS	WMTS
GetCapabilities	GetCapabilities
GetMap (CRS, BBOX,...)	GetTile (tileMatrixSet, tileMatrix, tileCol, tileRow, ...)
GetFeatureInfo	GetFeatureInfo

**Tabla 1.** Operaciones disponibles en los servicios WMS y WMTS de OGC.

Cada tesela de una matriz de teselas se identifica por el índice de columna (TileCol) y de fila (TileRow); estos índices tiene su origen 0,0 en la tesela izquierda y superior de la matriz y se incrementan hacia la derecha y hacia abajo respectivamente (ver figura 2).

En el WMTS, se definen tres maneras diferentes para obtener los recursos; dos de ellas están basadas en la obtención de los recursos a partir de operaciones, una siguiendo una sintaxis KVP (*Key-Value Pairs*) y otra siguiendo una sintaxis SOAP (*Simple Object Access Protocol*); la tercera está basada en la obtención de los recursos directamente a partir de su URL (*Uniform Resource Locator*) según el paradigma RESTful.

Actualmente el proveedor de servicios web puede escoger entre “WMS clásico” (obtener mapas con peticiones *GetMap* e indicando la envolvente completa de la ventana) y los “servicios basados en teselas” (ya sea a partir de WMS teselado o de un estándar específico para teselas como WMTS). En el siguiente apartado se expone cuándo los servicios basados en teselas son más adecuados y cómo diseñarlos y usarlos correctamente.

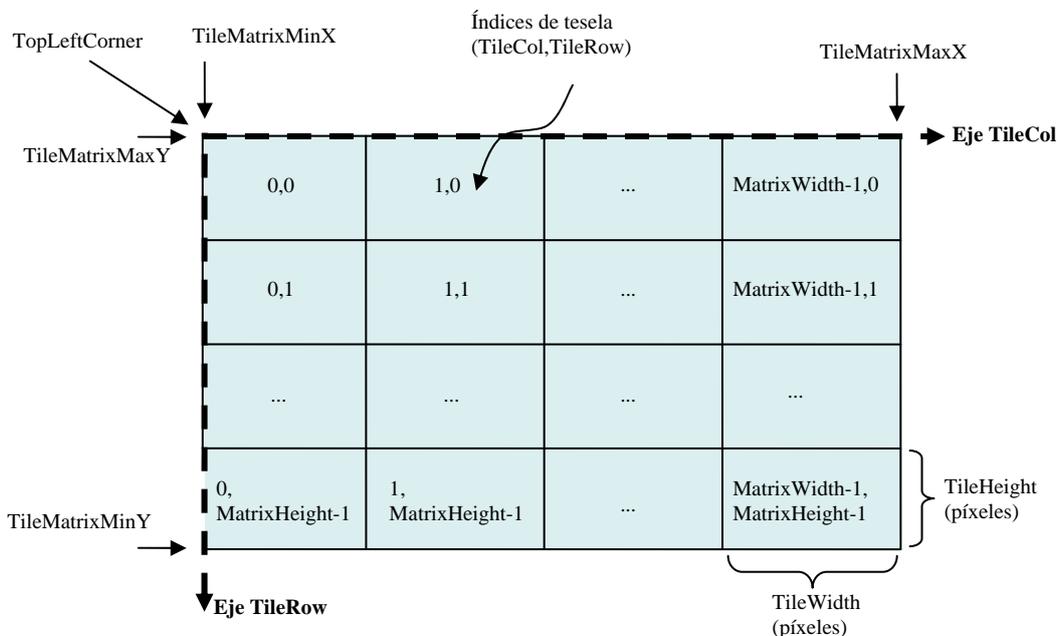


Figura 2. Espacio de teselas.

## DISCUSIÓN

Actualmente algunos clientes de servicios Web están usando los servicios WMS como si fueran sistemas basados en teselas (como en el caso de OpenLayers de OSGeo) (OpenLayers, 2010).

Las siguientes ecuaciones publicadas por Masó *et al.* (2010a) permiten usar un cliente basado en teselas contra un servidor, es decir, nos permiten convertir la pareja de índices de teselas (*tileCol*, *tileRow*) a un envolvente (*leftX*, *rightX*, *upperY*, *lowerY*) en un sistema de referencia determinado para poder formular una petición *GetMap* a un servicio WMS estándar.

$$\begin{aligned} \text{leftX} &= \text{tileCol} \times \text{tileSpanX} + \text{tileMatrixMinX} \\ \text{upperY} &= \text{tileMatrixMaxY} - \text{tileRow} \times \text{tileSpanY} \\ \text{rightX} &= (\text{tileCol} + 1) \times \text{tileSpanX} + \text{tileMatrixMinX} \\ \text{lowerY} &= \text{tileMatrixMaxY} + (\text{tileRow} + 1) \times \text{tileSpanY} \end{aligned}$$

$$\begin{aligned} \text{tileSpanX} &= \text{tileWidth} \times \text{pixelSpan} \\ \text{tileSpanY} &= \text{tileHeight} \times \text{pixelSpan} \\ \text{pixelSpan} &= \frac{\text{scaleDenominator} \times 0.28 \cdot 10^{-3}}{\text{metersPerUnit}} \end{aligned}$$

$$\begin{aligned} \text{tileMatrixMaxX} &= \text{tileMatrixMinX} + \text{tileSpanX} \times \text{matrixWidth} \\ \text{tileMatrixMinY} &= \text{tileMatrixMaxY} - \text{tileSpanY} \times \text{matrixHeight} \end{aligned}$$

El principal problema de este uso “teselado” del WMS reside en que el cliente no dispone de la información sobre cómo es la estructura de la pirámide de teselas. Un problema adicional aparece debido a que cada tesela puede identificarse de diversas formas según el orden de los parámetros del WMS o según el número de decimales indicados en las coordenadas de la envolvente de la tesela; lo que confunde a los sistemas de caché de Internet y provoca problemas de redondeo en las operaciones en punto flotante que pueden dar lugar a imágenes ligeramente distintas de las esperadas.

WMS-C (*Web Map Service-Cache*) es una adaptación del WMS editada por OSGeo que añade la descripción del conjunto de matrices de teselas a los metadatos del servicio WMS. Pero debido a que las peticiones se continúan realizando en el formato WMS los problemas de orden de los parámetros y de redondeo continúan existiendo (WMS Tile Caching, 2010).

Esencialmente WMTS proporciona una descripción del conjunto de la matriz de teselas a los metadatos del servicio y soluciona el problema de redondeo por punto flotante al usar identificadores en lugar de números decimales. WMTS aporta una sintaxis RESTful más acorde con el uso de Internet dado que cada tesela recibe una URL directa, definida a partir de una plantilla de URL (*URL template*) publicada en los metadatos del servicio, por lo que resulta más fácil de combinar en páginas Web junto con otros contenidos a partir de un vínculo.

TMS (*Tile Map Service*) es otro estándar de OSGeo que introduce el uso de índices de teselas y aporta una sintaxis RESTful, pero es una aproximación no armonizada con los estándares OGC y que requiere de diversas peticiones concatenadas para llegar a la URL de una tesela concreta, dificultando su uso (Tile Map Service specification, 2010).

En general existen dos importantes beneficios en el uso de los sistemas basados en teselas. Desde el punto de vista del servidor, las peticiones de las teselas pueden identificarse como iguales por los mecanismos de caché de Internet reduciendo el trabajo del servidor en operaciones repetitivas (Scholten, 2006) (salvando las diferencias antes expuestas en el WMS “teselado”). Y desde el punto de vista del cliente, los desplazamientos laterales y los redibujados de pantalla presentan una experiencia mejorada dado que gran parte de las teselas recuperadas en el redibujado anterior pueden reaprovecharse.

Es importante destacar que estos dos aspectos son independientes y ambos, por separado, proporcionan beneficios suficientes, aunque la combinación de ambos ha dado como resultado el éxito de los navegadores de mapas dirigidos al gran público como el *Google Maps* o *Microsoft Bing*. Además, el punto de vista del cliente cobra cada vez mayor importancia dado que, por un lado, ha aumentado el tamaño de las pantallas y las ventanas de navegación y el número de teselas que el cliente puede reusar y, por otro lado, porque las aplicaciones GIS profesionales están usando cada vez más los servicios Web como orígenes de datos cartográficos. Las aplicaciones GIS profesionales acostumbran a realizar más redibujados de la ventana dado que la cartografía proporcionada por los servicios Web debe combinarse con otro tipo de cartografía en local y que el usuario abre herramientas que cubre

partes de los datos temporalmente y se realizan cambios en la visualización sobre los datos que requieren de actualizaciones frecuentes de la pantalla.

En este sentido, Catastro (2010) indica que "la navegación sobre una capa teselada es muy eficaz [...] siempre que el servicio haya sido diseñado para recibir este tipo de peticiones. En caso contrario cada vez que un usuario accede a consultar la cartografía, el cliente solicita hasta 30 consultas simultáneas a base de datos de otros tantos de teselas de la cartografía. El resultado es que [aumenta] el número de consultas concurrentes [...] lo que supone una degradación del servicio". Esta afirmación sólo considera el punto de vista del servidor e ignora las ventajas innegables desde el punto de vista del cliente y la mejora de la experiencia del usuario.

Existen algunos malentendidos respecto al uso de los sistemas de mapas basados en teselas que vamos a desglosar progresivamente en esta discusión y que se detallan a continuación:

- Sólo pueden usarse para cartografía estática (que no necesita de actualizaciones frecuentes).
- Sólo permiten disponer de información prerasterizada.
- Son más rápidos que los sistemas basados en peticiones de ventana completa y la experiencia del usuario es mejor.

Si bien es cierto que los sistemas basados en teselas son más adecuados para cartografía estática esto no significa que no pueden usarse para cartografía dinámica. Cuando se trabaja con información estática es muy beneficioso que los sistemas basados en teselas realicen una preparación de las capas que consiste en la prerenderización de todas y cada una de las teselas en todos los sistemas de referencia y conjuntos de matrices de corte en los que el servidor quiera ofrecer la capa. Este procedimiento es completamente automatizable, por lo que puede repetirse tantas veces como sea necesario, ya sea total o parcialmente. Esto resulta especialmente útil si se prevé que la información cambie cada cierto tiempo (Yang et al., 2005). Aunque se realicen cambios frecuentes, generalmente éstos solo afectaran a una zona geográfica concreta, por lo que solo se requerirá la actualización de un número limitado de teselas.

En el caso de que la información sea totalmente dinámica y no sea posible mantenerla preparada en tiempo real, deberá renunciarse a este procedimiento de preprocesado, pero eso no significa que no pueda utilizarse un sistema de teselas; sólo significa que las teselas deben sintetizarse dinámicamente cuando se produzca cada petición.

Cuando la cartografía presentada tenga gran extensión y mucho detalle o resolución, raramente dos usuarios visitarán a la vez exactamente la misma zona en los niveles de zoom de máximo detalle. El espacio requerido en disco para la generación de todas las teselas, sobretodo de los niveles de zoom de mayor detalle, puede suponer un espacio excesivamente grande y un tiempo de generación excesivamente largo. Sin embargo, los niveles de poco detalle y los niveles de detalle medio en las zonas de mayor interés, serán más frecuentemente visitados (Quinn y Gahagen, 2010). En este caso se puede realizar un esquema mixto en el que los niveles de zoom con menor detalle se mantendrán prerasterizados mientras que los de mayor detalle se elaborarán dinámicamente.

Cuando la información cambia con el tiempo es muy importante informar de la caducidad de la información a los mecanismos de caché para evitar que las actualizaciones no sean entregadas al servidor y en su lugar se reciba una versión antigua des de un *proxy* intermedio. Las cabeceras HTTP permiten especificar la fecha en la que se prevé que el contenido caduque y de ese modo informar al servidor de caché de la obsolescencia de las teselas almacenadas.

Aunque nuestra experiencia frente a los servicios de mapas para el gran público parece demostrar que los sistemas basados en teselas son más rápidos que los sistemas basados en WMS clásico, en general, no tiene porque ser cierto. En particular, los servicios que realicen una prerasterización de las capas en teselas deberían de ser más rápidos que el WMS clásico; sin embargo, cuando por las razones antes expuestas no sea posible prerasterizar las teselas, y éstas deben generarse dinámicamente, la velocidad del servicio dependerá fuertemente de la implementación y de las estrategias de optimización que se realicen. En particular, Wang y Gong (2008) comparan el WMS clásico de MapServer con una implementación basada en teselas y determina que el primero es más rápido. Sin embargo, hay que discutir más a fondo estos resultados, objetivo que acometeremos a continuación.

La principal razón por la que un servicio de mapas basado en teselas puede no ser eficiente es la saturación de servicio. Cuando un cliente WMS clásico solicita una vista para su ventana realiza una única petición por ventana al servidor; el servidor procede a acceder a los datos y a rasterizar dinámicamente los datos necesarios, devolviendo el resultado. Por otro lado, los clientes basados en teselas desarrollados para navegadores Web, para rellenar la vista del mapa realizan varias peticiones al servidor; recibiendo éste, casi simultáneamente, estas peticiones y atendíéndolas de manera independiente; accediendo varias veces a los mismos datos, que se generalmente se

encuentran en el mismo soporte, y generando diversas rasterizaciones independientes que generalmente compiten por los mismos recursos de memoria y de procesador (Tu, 2004). Si el sistema no ha sido específicamente diseñado para esta situación, un solo cliente puede producir el colapso momentáneo del servidor. Unos pocos clientes basados en teselas conectados al mismo servidor producen un efecto similar a un ataque de denegación de servicio dejándolo incapacitado para atender a más clientes.

Por ello, estrategias de optimización orientadas a una mayor rapidez del servicio WMS clásico pueden no ser adecuadas o no ser suficientes para el caso de peticiones basadas en teselas. Cuando no es posible la prerasterización de los datos en el servidor, o incluso en este caso, es recomendable el uso de estrategias de optimización específicamente orientadas a los servicios basados en teselas.

Como se ha explicado anteriormente, los clientes basados en teselas tienden a generar diversas peticiones concurrentes al servidor. Una primera estrategia, que permitiría solucionar este problema, es limitar el número de peticiones concurrentes en las capacidades del servidor, típicamente a partir de la configuración del servidor Web (*Internet Information Server, Apache, etc.*).

Otra estrategia posible, es mejorar el hardware del servicio, sobretodo cuando no es posible actualizar el software de servicio, por ejemplo, por que no se tiene acceso al código fuente; adquiriendo un hardware de mayores prestaciones. Sin embargo, actualmente, la fabricación de ordenadores servidores de altas prestaciones se encuentra limitada por factores tecnológicos, y al mismo tiempo tienen un precio elevado por lo que la relación precio/beneficio es baja. Afortunadamente existen otras soluciones más asequibles y que permiten una mejor escalabilidad basadas en clusters de servidores. En nuestro caso, la mejor solución analizada es la arquitectura de balanceo de carga de red (*Network Load Balance, NLB*) que consiste en distribuir el trabajo a realizar (carga) entre dos o más ordenadores; el balanceo de la carga del sistema se realiza a partir de un algoritmo que distribuye la carga entre los ordenadores; los balanceos más simples lo realizan aleatoriamente o siguiendo una planificación *round-robin* en la que se seleccionan todos los ordenadores de forma equitativa y siguiendo un orden racional. Cada uno de los servidores que forma el cluster puede ser un ordenador de uso común por lo que el sistema resulta muy económico.

A parte de su bajo coste y fácil implementación, las principales ventajas de la arquitectura NLB son la facilidad de incorporar nuevos equipos al cluster en caso de necesidad y que no existe un ordenador central de coordinación por lo que el sistema permite retirar cualquier ordenador averiado del grupo sin necesidad de reconfiguración. La principal desventaja de este sistema es la necesidad de replicación de los datos en todas las máquinas para conseguir el máximo rendimiento y que, sobretodo en los sistemas simples y más usados, no se considera la carga de trabajo de cada máquina por lo que, en el caso poco probable de que un ordenador se halle colapsado, continuará recibiendo peticiones.

Si se dispone del código fuente del servicio o se está desarrollando una implementación nueva del servicio entonces es posible seguir otra estrategia optimizando el software. El principal inconveniente de la programación en entornos web es que cada petición se ejecuta de manera independiente de las demás. Por ello, muchas implementaciones recargan del disco la información necesaria para generar un mapa cada vez que se recibe una petición. Para el caso de los servidores de teselas dinámicas, esta aproximación es particularmente penalizadora. Una alternativa es que la aplicación servidora dialogue con una aplicación que se ejecute permanentemente en el servidor. De este modo, cada petición de tesela iniciará una aplicación que solo pedirá a la aplicación permanentemente en ejecución la generación de la tesela y esperará. La aplicación en ejecución permanente ya tendrá cargada en memoria la información necesaria para generar la tesela, o cuanto menos, una parte de ella, por lo que su respuesta será sensiblemente más rápida.

Otra estrategia de optimización de software, que da buenos resultados, es la generación de metateselas (*metatiles*). Un metatesela es una agrupación en el servidor de un cierto número de teselas (por ejemplo la unión de 4x4 teselas contiguas en una sola imagen). La idea consiste en suponer que las peticiones de teselas no vienen nunca solas y que, recibida una petición de una tesela concreta, es muy probable que el servidor deba atender peticiones de las teselas adjuntas. En lugar de generar la tesela solicitada, se genera la metatesela completa, se almacena temporalmente y se extrae la tesela solicitada. Para la siguiente petición recibida, se comprobará si forma parte de alguna de las metateselas que se han generado hace poco y, en este caso, la información se extraerá de la metatesela correspondiente sin que sea necesario recuperar la información original, con lo que se reduce considerablemente el esfuerzo requerido para este servidor. Además, la generación de la metatesela garantiza que las teselas que la forman han sido generadas a partir de la misma información y de una manera más homogénea (como por ejemplo, con los mismos criterios de etiquetado de elementos reduciendo el riesgo de dobles etiquetas en la pantalla del usuario).

## CASO DE USO

Los autores de esta comunicación han implementado en el entorno MiraMon con éxito una aproximación por prerasterizado completo que utilizan para servir teselas WMTS y también para responder peticiones WMS convencionales. El servidor WMS genera el mapa a partir del nivel de zoom más próximo de la pirámide interna e interpola la imagen para ajustarla exactamente a la petición de la envolvente recibida.

Para garantizar el correcto funcionamiento del sistema en situaciones de elevada carga, se han adquirido 6 ordenadores convencionales de sobremesa de características Intel Quad Core, montadas sobre placas ASUS, con 4Gb de memoria y 3Tb de disco. La única particularidad es que han sido montados en cajas para armario (rack 19"/42U) para ahorrar espacio y permitir futura ampliación del sistema, que se les ha dotado de más espacio de disco duro y que disponen de doble tarjeta de red (ver figura 3). El sistema operativo instalado ha sido Windows 2003 Advanced Server que permite la configuración en red de NLB con dirección IP única para la red exterior y una red interior de alta velocidad que permite la sincronización rápida del contenido de los equipos.

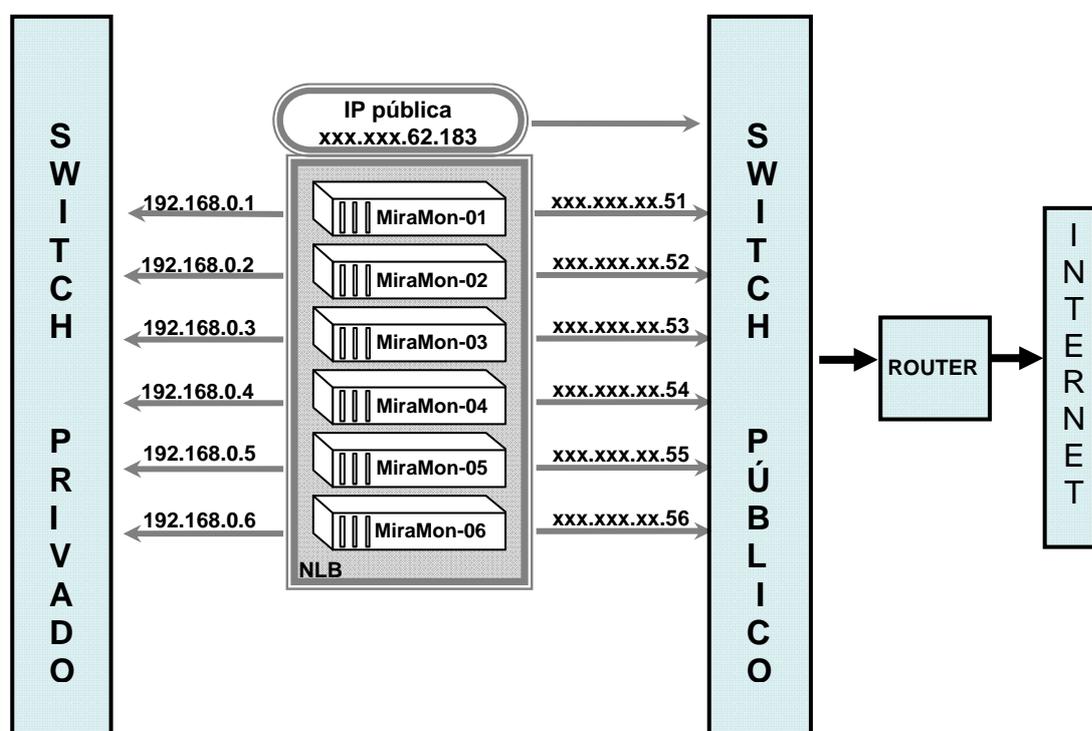


Figura 3: Esquema del sistema NLB implementado

Anteriormente se había utilizado 3 ordenadores dual Xeon de alta gama con servicios de mapas distintos que presentaba problemas puntuales de saturación. El cluster NLB actual concentra todos los servicios en una sola IP (*Internet Protocol*) y se ha demostrado mucho más eficiente, repartiendo la carga entre los 6 ordenadores por igual evitando la saturación de los mismos siendo esta solución más económica que la anterior y presentando la posibilidad de incluir fácilmente nuevos ordenadores al cluster en caso de incremento de la demanda.

## CONCLUSIONES

Los servidores de mapas que deben atender peticiones basadas en teselas deben diseñarse de manera distinta que los servidores que solo pretenden atender peticiones WMS clásico. Desde la aparición de las estrategias de teselas que usan sintaxis WMS, como OpenLayers, todos los servicios pueden recibir peticiones basadas en teselas por lo que el problema afecta a todos los servidores de mapas en general.

El principal problema de los servidores de teselas es que reciben diversas peticiones casi simultáneas, por lo que son más sensibles a la saturación de servicio, incluso con un número reducido de usuarios. La principal estrategia de hardware para la mejora del rendimiento es la paralelización de las peticiones, distribuyéndolas entre diversos servidores, tal como se consigue con la arquitectura de balanceo de carga NLB. Una de las estrategias más efectivas por software es el prerasterizado. Cuando esto no es posible, debido al elevado volumen de información o a una actualización constante de los datos, estrategias basadas en metateselas temporales permiten responder a varias peticiones individuales con una carga similar a la que produce un cliente basado en WMS clásico.

Los servidores de MiraMon han sido probados en una arquitectura de balanceo de carga NLB, demostrando que el rendimiento de los servicios de mapas aumenta considerablemente con un mínimo coste adicional respecto a otras estrategias y permitiendo ampliar la capacidad del servicio en un futuro cuando aumente la carga.

## REFERENCIAS BIBLIOGRÁFICAS

Catastro (2010) Peticiones teseladas al servicio WMS de la Dirección General del Catastro. En: [http://www.catastro.meh.es/esp/servicios\\_destacados1.asp#menu5](http://www.catastro.meh.es/esp/servicios_destacados1.asp#menu5)

De la Beaujardiere, J. (2004): *OGC Web Map Service (WMS) Interface*. Ver.1.3.0, OGC 03-109r1. En: <http://www.opengeospatial.org/standards/wms>

Masó J., and Julià N. (2008): Historia y estado actual del futuro estándar Web Map Tiling Service del OGC. *Jornadas Técnicas de la IDE de España 2008 (JIDEE2008)*. Tenerife.

Masó J., Pomakis K., Julià N. (2010a): *OGC Web Map Tile Service (WMTS) Implementation Standard*. Ver 1.0, OGC 07-057r7. En: <http://www.opengeospatial.org/standards/wmts>

Masó J., Zabala A., Julià N. (2010b): Fast Satellite Data Dissemination of Large JPEG2000 Compressed Images with OGC Standards. *Geomatics for Crisis Management 2010 (Gi4DM20710)*. Torino.

OpenLayers project in OSGeo (2010). En: <http://openlayers.org/>

Quinn S, Gahegan M (2010) *A Predictive Model for Frequently Viewed Tiles in a Web Map Transactions in GIS*, 2010, 14(2): 193–216.

Scholten M., Klamma R. and Kiehle C., (2006): Evaluating Performance in Spatial Data Infrastructures for Geoprocessing. *IEEE Internet Computing*, October 2006, pp. 34–41.

Tile Map Service Specification in OSGeo (2010). En: [http://wiki.osgeo.org/wiki/Tile\\_Map\\_Service\\_Specification](http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification)

Tu S, Flanagan M, Wu Y, Abdelguerfi M, Normand E, Mahadevan V, Ratcliff J, Shaw K (2004): Design Strategies to Improve Performance of GIS Web Services. *Information Technology; Coding and Computing*, 2: pp. 444-450.

Wang T., Gong J (2008) *Comparison of Tile Server Design Approaches for 3-D Geo-Visualization The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B2. Beijing.

WMS Tile Caching in OSGeo (2010). En: [http://wiki.osgeo.org/wiki/WMS\\_Tile\\_Caching](http://wiki.osgeo.org/wiki/WMS_Tile_Caching)

Yang (Phil), Chaowei, Wong, David W., Yang, Ruixin, Kafatos, Menas and Li, Qi (2005): Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science*, 19(3), pp.319- 342.