

Pruebas benchmark de soluciones cliente/servidor en software libre

Esbrí Palomares, Miguel Ángel¹
Higón Valero, José Vicente²

Departamento de Lenguajes y Sistemas Informáticos, Universidad Jaume I (Castellón), mesbri@uji.es¹
Conselleria de Infraestructuras y Transportes de la Generalitat Valenciana (Valencia), higon_jos@gva.es²

Resumen: *Presentamos los primeros resultados obtenidos de una evaluación de implementaciones basadas en software libre de los componentes IDE, centrando nuestro análisis en los servicios de visualización de mapas (Web Map Service), de acceso y manipulación de objetos geográficos (Web Feature Service) y repositorios de información geográfica (bases de datos e información vectorial). El objetivo es valorar de forma objetiva el estado de madurez de algunas de las soluciones abiertas más comunes, revelando sus fortalezas y deficiencias bajo una variedad de condiciones de número de usuarios, carga de datos, sistema operativo, licencias de uso, portabilidad, tiempos de respuesta, etc. Estas pruebas benchmark una vez completadas se publicarán enteras como una contribución más a una IDEE abierta y heterogénea respecto a su implementación, sin que este estudio pretenda menoscabar las soluciones propietarias ya existentes en el mercado.*

INTRODUCCIÓN

El objetivo de este trabajo es valorar de forma objetiva el estado de madurez de las soluciones *open source* que implementan los distintos componentes que conforman la arquitectura de las IDEs [3], revelando sus fortalezas y deficiencias bajo una variedad de condiciones como el número de usuarios, la carga de datos, el sistema operativo, licencias de uso, portabilidad, tiempos de respuesta, lenguaje de programación utilizado para su implementación, etc.

Hasta la formación del Open Geospatial Consortium, debido a la limitada funcionalidad proporcionada por los SIG tradicionales y para conseguir un nivel básico en la compatibilidad de los datos disponibles, las organizaciones con información espacial en múltiples sistemas y formatos tenían que crear aplicaciones contra APIs propietarias. Esto requería de un conocimiento profundo de los sistemas computacionales subyacentes, llegando al caso de tener que desarrollar n aplicaciones para acceder y/o manipular n formatos distintos de datos. Los datos se replicaban y transformaban continuamente según las necesidades del momento, socavando la calidad e integridad de las fuentes de datos espaciales. Como consecuencia de todo esto, la utilización de los sistemas propietarios cerrados se habían convertido en la forma habitual de trabajo, ocurriendo que el software desarrollado por una empresa podía o no funcionar con otros productos propios de su línea, pero con certeza no lo haría con otra solución propietaria distinta si no existía una relación explícita entre ambos que lo posibilitara.

A través de las Especificaciones de Implementación OpenGIS, especialmente las del conjunto OGC Web Services (OWS) [7], y junto con el modelo de referencia de la arquitectura de las IDEs definido según la NASA [8], FGDC [9] e INSPIRE [2], las distintas empresas y administraciones pueden dar acceso tanto a nivel global como local de forma transparente al usuario final (ya sea administración, empresas y organizaciones tanto a nivel europeo, nacional o local, así como a ciudadanos individuales) a servicios integrados de información espacial [1]. Esto permite la aparición de nuevas aplicaciones interoperables con ellos a través de estas interfaces estándar, posibilitando la eliminación de la redundancia o duplicación de datos geoespaciales y los conflictos que se dan entre las múltiples soluciones SIG existentes en el mercado, el acceso a través de la Web a los recursos espaciales, el e-comercio y el acceso seguro a estos. Además, en la práctica, la implementación de una especificación significa que los productos que son conformantes a ella pueden “comunicarse inteligentemente” con otros que también lo son, lo que permite cambiar en un momento dado uno de los componentes por otro de similares características, ya sea por motivos económicos (pago de licencias), de funcionalidad o rendimiento.

Dentro del modelo de referencia de la arquitectura de las IDEs pueden distinguirse cuatro grupos de componentes: aplicaciones de usuario (ej. visualización de diferentes capas de información geográfica provenientes de diferentes fuentes o recursos de información), servicios de geo-procesamiento (ej. análisis temporal y espacial, planificación, etc.; o el resultado de la encadenación de varios de estos servicios de geo-procesamiento [10]), servicios de catálogos (ej. publicación y/o descubrimiento de servicios o datos geoespaciales) y repositorios de información geográfica.

La comunidad de usuarios y desarrolladores de estos componentes no se ha mostrado ajena a la corriente del software libre, también común en otros ámbitos, y en los últimos años ha sido muy activa, desarrollando sus propias implementaciones para conseguir la independencia tecnológica frente a las soluciones propietarias ya existentes de estos componentes, bien sea por la necesidad de una solución de bajo coste o porque las ya existentes no se adaptan a sus necesidades.

En este trabajo se presentan los primeros resultados obtenidos de una evaluación de implementaciones de los componentes IDE basadas en software libre, centrandó el análisis en los servicios de visualización de mapas (*Web Map Service*), de acceso y manipulación de objetos geográficos (*Web Feature Service*) y repositorios de información geográfica (bases de datos espaciales e información vectorial). Los resultados obtenidos a partir de este análisis pretenden comparar cuantitativa y cualitativamente algunas de las soluciones abiertas más comunes. Estas pruebas benchmark una vez completadas se publicarán enteras como una contribución más a una IDEE abierta y heterogénea respecto a su implementación, sin que este estudio pretenda menoscabar las soluciones propietarias ya existentes en el mercado.

EL CONTEXTO DEL BENCHMARKING

A lo largo de la historia de la informática, el uso de pruebas en detalle o *benchmarks* ha sido algo muy común, de forma que los resultados obtenidos de forma objetiva en las distintas arquitecturas mediante estas pruebas podían ser comparados (ej. BRL-CAD, LINPACK, 3DMark, SPEC CPU2000, SPEC WEB99, etc.). En general, la realización de pruebas comparativas no suele ser una tarea fácil y requiere de sesiones repetitivas para llegar a conclusiones útiles, siendo también difícil la interpretación de los resultados de las pruebas. Un factor a tener en cuenta durante la realización de las pruebas es que los vendedores suelen afinar sus productos específicamente para los benchmarks más comúnmente utilizados en su sector, por lo que hay que tener especial precaución a la hora de interpretar los resultados. Además, los benchmarks generalmente, a parte de las mediciones cuantitativas del rendimiento de un sistema, no suelen tener en cuenta ninguna medición cualitativa a cerca del servicio como pueden ser la seguridad, la disponibilidad, la confiabilidad, la escalabilidad, o el grado de conformidad con las especificaciones, las cuales son tan o más importantes que las anteriores (ej. pruebas de conformidad de servicios WMS y WFS desarrolladas por OCCAMLAB [4] y el proyecto ACE-GIS [5]).

Dentro del amplio espectro de pruebas de benchmarking que existen para evaluar el rendimiento de casi cualquier componente software o hardware, hay dos campos que son de especial relevancia para el caso que nos ocupa en este trabajo: las bases de datos y los servicios web. Dentro de los objetivos de este trabajo no se encuentra el hacer un estudio en profundidad del rendimiento de los distintos gestores de bases de datos, ya sean libre distribución como MySQL y PostgreSQL o propietarias como Oracle Server y Microsoft SQL Server. Pero es interesante hacer mención de la existencia de multitud benchmarks orientados a la evaluación de estos, ya que las bases de datos, y en concreto aquellas que disponen de extensión espacial, son comúnmente utilizadas como repositorios para el manejo y acceso a la información geoespacial, bien sea de forma directa a través de clientes o indirectamente por medio de servicios web de geo-procesamiento como es el caso de los *Web Map Services* o los *Web Feature Services*.

Algunos de los benchmarks más utilizados para la evaluación de gestores de bases de datos son:

- **TPC Benchmark™ C (TPC-C)**, benchmark propietario utilizado para el procesamiento online de transacciones (en inglés, OLTP). Es considerado como el estándar de facto para OLTP en cuanto a benchmarks para gestores de bases de datos, principalmente porque intenta simular el procesamiento de transacciones online que se dan en el mundo real. <http://www.tpc.org>
- **eWeek's Nile benchmark**, diseñado para comparar los resultados de varios gestores de bases de datos (entre otros MySQL, Microsoft SQL Server, IBM DB2, Oracle y Sybase) en aplicaciones de comercio electrónico en el "mundo real". <http://www.eweek.com/article2/0,4149,293,00.asp>
- **SysBench**, benchmark open-source, modular, multi-plataforma y multi-hilo diseñado para evaluar los parámetros de los sistemas operativos que son importantes para un sistema que esté utilizando un gestor de bases de datos bajo una carga intensiva. <http://sysbench.sourceforge.net>
- Otros: **MySQL Benchmark Suite** (<http://dev.mysql.com/doc/mysql/en/mysql-benchmarks.html>), **The Open Source Database Benchmark (OSDB)**, (<http://osdb.sourceforge.net>), **BenchW** (<http://benchw.sourceforge.net>) y **Open Source Development Labs' (OSDL) Database Test Suite (DBT)** (http://www.osdl.org/lab_activities/kernel_testing/osdl_database_test_suite).

Por otro lado, los servicios web se están convirtiendo en una de las principales tecnologías para la comunicación entre las aplicaciones de distintas empresas, permitiendo a los clientes software intercambiar información con estas de una forma transparente mediante estándares y protocolos de transporte abiertos basados en XML. Dada la rápida implantación de los servicios web como estándar de facto para la transmisión y almacenamiento de información, es

necesario evaluar las características y rendimiento de las implementaciones de esta tecnología en los productos que lo ofrezcan. Algunos de los programas más utilizados para la evaluación de servicios web son:

- **ApacheBench (ab)**, herramienta de libre distribución diseñada para evaluar servidores HTTP de Apache. En particular muestra cuantas peticiones por Segundo es capaz de servir (viene incluida con el servidor Apache). <http://httpd.apache.org/docs/2.0/programs/ab.html>
- **Microsoft Web Application Stress Tool (WAST)**, herramienta gratuita diseñada para simular de forma realista múltiples navegadores pidiendo páginas de un servidor web, permitiendo recopilar información a cerca del rendimiento y estabilidad de la aplicación web. <http://www.microsoft.com/downloads/details.aspx?FamilyID=E2C0585A-062A-439E-A67D-75A89AA36495&displaylang=en>
- **LoadRunner**, herramienta de pago para la evaluación del comportamiento y rendimiento de aplicaciones web. Puede simular miles de usuarios y utiliza monitores de rendimiento para identificar y aislar problemas. Evaluación de servidores de aplicaciones web, servidores de streaming, gestores de bases de datos, aplicaciones Java y ERP. <http://www-svca.mercuryinteractive.com/products/loadrunner>
- **Apache Jmeter**, aplicación de libre distribución desarrollada completamente en Java. Inicialmente diseñada para medir el rendimiento de aplicaciones web, se extendió a otras funciones como servidores FTP, aplicaciones Java, SOAP/XML-RPC y JDBC. <http://jakarta.apache.org/jmeter/index.html>
- **TPC Benchmark™ W (TPC-W)**, benchmark transaccional para servicios web. La carga de trabajo se realiza en un ambiente de comercio electrónico controlado que simula las actividades de un negocio orientado a servidores web transaccionales (es decir, e-comercio). <http://www.tpc.org/tpcw/default.asp>
- Otros: **httperf** (<http://www.hpl.hp.com/research/linux/httperf>), Webstone (<http://www.mindcraft.com/benchmarks/webstone>), y Flood (<http://httpd.apache.org/test/flood>).

METODOLOGÍA DE EVALUACIÓN

El objetivo de esta sección es presentar la metodología utilizada en los análisis de los servicios de visualización de mapas (*Web Map Service*), de acceso y manipulación de objetos geográficos (*Web Feature Service*) y repositorios de información geográfica (bases de datos e información vectorial), sí como las soluciones *open source* y parámetros elegidos para el estudio.

Una vez concluido este trabajo, los resultados obtenidos también han de servir como orientación a la Consellería de Infraestructuras y Transportes (CIT) para tomar decisiones referentes a qué implementaciones de los distintos componentes utilizar en su IDE basada en tecnologías de software libre.

Servicios web de visualización y geo-procesamiento

Los *Web Map Service* proveen al cliente software de una forma estándar de acceder a través de la Web a mapas geo-referenciados en un formato de imagen, y permite a los proveedores ofrecer un catálogo de sus productos disponibles. Estos servicios proveen únicamente del nivel más básico de web mapping. Sus principales características son la construcción dinámica de un mapa como una imagen, como una serie de elementos gráficos o como un conjunto empaquetado de elementos geográficos; responder a consultas sencillas sobre el contenido del mapa; e informar a otros programas a cerca de los mapas que puede generar y a cuales de aquellos se les pueden hacer consultas sobre su contenido. Por otro lado, en el caso de los *Web Feature Service*, el servicio básico (No-Transaccional) da únicamente acceso de lectura a datos vectoriales utilizando GML (codificación XML para el transporte y almacenamiento de información geográfica, incluyendo tanto propiedades espaciales como no espaciales de los elementos geográficos) como el protocolo subyacente para realizar consultas espaciales, recuperar los resultados y manipular la geometría. El servicio Transaccional añade funcionalidades de bloqueo y actualización de los elementos geográficos al servidor para crear una nueva instancia geográfica, borrar y/o actualizar una instancia ya existente, y obtener o consultar elementos geográficos basados en restricciones espaciales y no espaciales.

Las soluciones elegidas para ser objeto de la evaluación son Minnesota MapServer (<http://mapserver.gis.umn.edu>, con licencia que permite que sea distribuido y modificado con total libertad) y Geoserver (<http://geoserver.sourceforge.net>, distribuido bajo licencia GPL). Ambas soluciones integran en una única aplicación los servicios de WMS y WFS, y pueden ser instaladas en servidores con sistemas operativos Windows y UNIX/Linux. La primera ha sido desarrollada en C++, mientras que la segunda en Java, contando las dos con una amplia comunidad de usuarios y desarrolladores.

Descripción de la organización del sistema

En los entornos que se describen a continuación, a parte del gestor de bases de datos elegido para el almacenamiento y consulta de la información geoespacial, también se ha optado por incluir en cada uno de los servidores una copia local de los ficheros vectoriales para poder evaluar el rendimiento de repositorios heterogéneos de información geoespacial

Entorno de las pruebas para los Web Feature Services

El sistema distribuido de geo-servicios utilizado para evaluar las prestaciones de las soluciones *Web Feature Service* implementadas se encuentra alojado en el Departamento de Lenguajes y Sistemas Informáticos de la Universidad Jaume I. Este sistema consiste en 4 nodos que actúan como servidores y 1 nodo desde el cual se realizarán las pruebas. Uno de los servidores aloja el gestor de la base de datos geoespacial y los tres servidores restantes alojan los servicios de geo-procesamiento.

El servidor del gestor de la base de datos (servidor 1) es un Intel Pentium IV XEON Dual a 1.5 GHz con 1 GB de memoria RAM, dos discos SCSI (17 GB + 37 GB) y sistema operativo Suse Linux 9.2 Professional. Funcionando sobre este está el gestor de base de datos PostgreSQL/PostGIS. Las versiones del software utilizado para la base de datos espacial son PostgreSQL 8.0.3 y PostGIS 1.0.4, incluyendo las librerías necesarias para el manejo de geometría y proyecciones Geos 2.1.4 y Proj 4.4.9 respectivamente.

Los servidores que alojan el geo-servicio son un Intel Pentium IV XEON a 2.4 GHz con 512 MB de memoria RAM, disco duro IDE de 80GB y sistema operativo Windows XP Professional (Service Pack 2); un Intel Pentium IV a 1.8 GHz con 512 MB de memoria RAM, disco duro IDE de 80GB y sistema operativo Suse Linux 9.2 Professional; y un Intel Pentium IV XEON Dual a 3.0 GHz con 2 GB de memoria RAM, dos discos SCSI de 136 GB y sistema operativo Suse Linux 9.2 Professional. Todos los servidores llevan instalada la máquina virtual de Java (JDK 1.4.2_08) y el contenedor de servlets Apache Tomcat (versión 5.0.28) y las implementaciones *open source* de los *Web Feature Service*, Geoserver (versión 1.3.0-beta3) y el Minnesota Mapserver (versión 4.6.1) El servidor Windows utiliza el paquete MS4W (<http://www.maptools.org/ms4w/index.phtml>, versión 1.2.1) que incluye el Minnesota Mapserver ya compilado para este sistema operativo. Este paquete incluye además el servidor web Apache (versión 2.0.54), y las librerías necesarias para el manejo de proyecciones (proj, versión 4.4.6 y formatos vectoriales y acceso a PostGIS (GDAL/OGR, versión 1.2.6). Por otro lado, los servidores Linux utilizan el servidor web Apache (versión 2.0.50) y el Minnesota Mapserver ha sido compilado a partir del código fuente, junto con todas las librerías necesarias (proj4 versión 4.4.9, GDAL/OGR versión 1.2.6, geos versión 2.1.1 y PostGIS versión 1.04).

Estos servidores los llamaremos servidor 2, 3 y 4 respectivamente para que puedan ser referenciados posteriormente cuando hablemos de los resultados.

El nodo desde el cual se lanzan las pruebas es un Intel Pentium IV a 3.0 GHz con 1 GB de memoria RAM, dos discos duros SERIAL ATA de 120 GB y sistema operativo Windows XP Professional (Service Pack 2).

Todos los nodos del sistema disponen de una tarjeta de red ethernet 100MB/s y están conectados a la red local por medio de un switch con puertos de velocidad 10, 100 y 1000 MB/s.

Entorno de las pruebas para los Web Map Service

El sistema distribuido de geo-servicios utilizado para evaluar las prestaciones de las soluciones *Web Map Service* implementadas se encuentra alojado en la Consellería de Infraestructuras y Transportes de la Generalitat Valenciana. Este sistema consiste en 2 nodos que actúan como servidores y 1 nodo desde el cual se realizarán las pruebas. Ambos servidores alojan los distintos repositorios de información geoespacial (gestor de base de datos y ficheros vectoriales) y los servicios de visualización.

Los servidores son un Intel Pentium IV a 2.8 GHz con 512 MB de memoria RAM, disco duro IDE de 80GB y sistemas operativos Windows XP Professional y Suse Linux 9.0 Professional; y un Intel Pentium IV Cuádruple a 3.0 GHz con 4 GB de memoria RAM, cinco discos SCSI de 136 GB y sistema operativo Suse Linux Enterprise. Ambos servidores han sido configurados con el mismo software. Las versiones del software utilizado para la base de datos espacial son PostgreSQL 8.0.4 y PostGIS 1.0.4, incluyendo las librerías necesarias para el manejo de geometría y proyecciones Geos 2.1.4 y Proj 4.4.9 respectivamente. La versión de la máquina virtual de Java utilizada es JDK 1.5_05, el contenedor de servlets es Apache Tomcat (versión 5.5.12) y el servidor web es Apache (versión 2.0.54). Las implementaciones *open*

source de los *Web Map Service* son Geoserver (versión 1.2.4) y el Minnesota Mapserver (versión 4.6.1), este último con todas las librerías necesarias para acceder a los repositorios raster y vectoriales (proj4 versión 4.4.9, GDAL/OGR versión 1.3.0, geos versión 2.1.1 y PostGIS versión 1.04).

El nodo desde el cual se lanzan las pruebas es un ordenador portátil Intel Pentium M a 2.0 GHz con 1 GB de memoria RAM y sistema operativo Suse Linux 9.2 Professional.

Todos los nodos del sistema disponen de una tarjeta de red ethernet 100MB/s y están conectados a la red local por medio de un switch con puertos de velocidad 10, 100 y 1000 MB/s.

Descripción de los datos de prueba

Para la realización de las pruebas se han utilizado tres conjuntos de datos espaciales de gran tamaño obtenidos de la Consellería de Infraestructuras y Transportes de la Generalitat Valenciana: Usos del suelo (capa con geometría poligonal, escala 1:25000), Red de comunicaciones (capa con geometría lineal, 1:10000) y una capa con geometría puntual a escala 1:25000 obtenida a partir de la capa de polígonos. Este conjunto de datos ha sido manipulado para tener un cierto control sobre el número de elementos geográficos que contiene cada capa. Para cada tipo de geometría simple (punto, línea y polígono) se han obtenido tres capas de distintos tamaños (puntos_p, puntos_m, puntos_g, líneas_p, líneas_m, líneas_g, poligon_p, poligon_m y poligon_g). El tamaño de estos datos es de aproximadamente 289 KB, 2.79 MB y 27.9 MB para los puntos; 464 KB, 10.6 MB y 140 MB para las líneas; y de 2.78 MB, 16.7 MB y 134 MB para los polígonos en el formato comprimido Shapefile de ESRI. Una vez importados a la base de datos PostgreSQL, el número de registros es de 200, 2000 y 20000 en el caso de los puntos; 2500, 25000 y 250000 en el caso de las líneas; y 5000, 50000 y 500000 en el caso de los polígonos. Las tablas de la base de datos hacen uso de índices espaciales para acelerar las consultas.

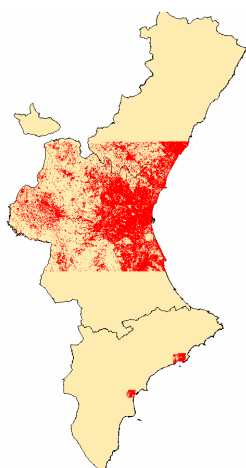


Figura 1: Capa de puntos original

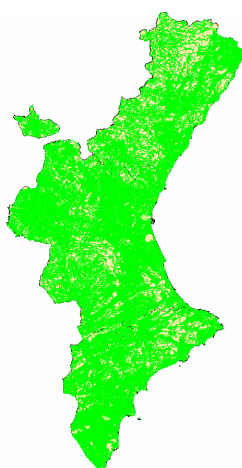


Figura 2: Capa de líneas original

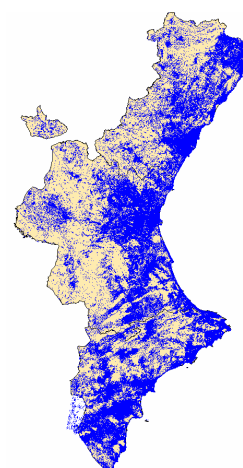


Figura 3: Capa de polígonos original

Métrica utilizada

Aunque se pretende evaluar geo-servicios que ofrecen funcionalidades distintas (WMS y WFS), hay una serie de cuestiones de carácter general que son aplicables a ambos, como qué operaciones de las especificaciones de implementación definida por OGC han sido implementadas en cada una de las soluciones, cuál es el tiempo de respuesta de cada una de las soluciones ante las peticiones recibidas, qué sistema operativo es el adecuado, cuál es el efecto del hardware en las prestaciones en las distintas soluciones, cómo se comportan las distintas soluciones a medida que se incrementa la carga de usuarios o qué impacto tiene el uso de las distintas fuentes de información geoespacial (base de datos versus ficheros locales) en el rendimiento. Por el contrario, otras cuestiones son específicas del geo-servicio en cuestión.

En el caso del *Web Map Service* es interesante plantearse con qué formatos raster trabaja más eficientemente cada servidor o cómo afectan al rendimiento de estos la utilización de estilos (Ej., leyendas, simbología, etc.) o reproyecciones de las capas; mientras que en el caso de los *Web Feature Services* sería interesante ver cómo afectaría al rendimiento y al tiempo de respuesta si se aplicasen métodos de compresión al mensaje de respuesta. Si bien muchos de

estos parámetros tienen una respuesta numérica como el tiempo de respuesta en segundos, el porcentaje de peticiones fallidas, o el número de respuestas por segundo; otros como pueden ser el grado de conformidad con las especificaciones requieren de una respuesta cualitativa.

En las soluciones que implementan el *Web Feature Service*, se aplicarán sobre cada una de las capas que ofrece el servidor cuatro operaciones típicas definidas en la especificación [6], GetFeature sin filtros (se obtiene todos los elementos geográficos de la capa), GetFeature con filtro espacial DWithin (radio de búsqueda de 100km), GetFeature con filtro espacial BBOX y GetFeature con filtro espacial Intersects, (el área utilizada en las dos últimas operaciones es de 36.77 millones de km²).

En las soluciones que implementan el *Web Map Service*, se aplicará sobre cada una de las capas que ofrece el servidor la operación GetMap definida en la especificación [11].

La herramienta utilizada para la realización de las pruebas es JMeter. Para simular una carga de 1, 5 ó 10 usuarios en cada servidor se lanzará simultáneamente la misma operación a las distintas capas de puntos, líneas y polígonos configuradas en los servidores tantas veces como usuarios se quiera simular dejando un intervalo de 2 segundos entre cada petición, repitiendo este proceso 30 veces para sacar un tiempo medio de respuesta. Además, también se han tenido en cuenta dos tiempos para la realización de las pruebas, un time out de 30 segundos por cada petición lanzada, el cual una vez transcurrido sin una respuesta por parte del servidor se considera que esa petición ha fallado y un tiempo máximo de 120 segundos para que el servidor mande el mensaje completo de respuesta.

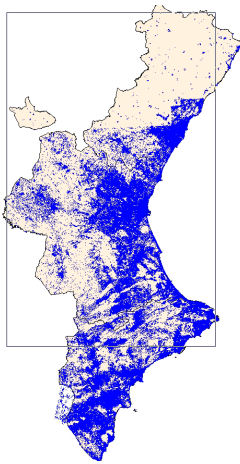


Figura 4: BoundingBox utilizada en operación espacial BBOX sobre la capa poligon_g (WFS)

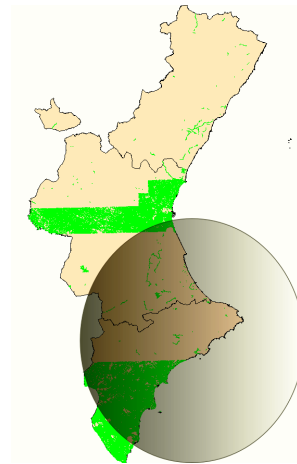


Figura 5: Radio de búsqueda utilizado en la operación espacial DWithin sobre la capa lineas_g (WFS)

RESULTADOS PRELIMINARES

Antes de iniciar las pruebas sobre las soluciones escogidas, se realizó un estudio preliminar de las capacidades ofrecidas por cada una de ellas. Como se ha comentado anteriormente, Mapserver y Geoserver disponen de una amplia comunidad de usuarios que informan de los errores hallados, lo cual hace de ellas unas soluciones muy probadas y confiables. Mapserver utiliza C++ como lenguaje para su implementación y Geoserver el lenguaje Java, previendo en una primera aproximación que la primera solución será la más rápida. Por contra, Geoserver es multiplataforma y puede ser utilizado de forma inmediata en cualquier sistema sin que se requiera de grandes conocimientos informáticos por parte del usuario, mientras que Mapserver y las librerías que utiliza necesitan ser compilados específicamente para la arquitectura del sistema en el que vayan a ser utilizados, convirtiéndose en una tarea de elevada complejidad para los usuarios noveles.

Mapserver implementa únicamente las operaciones no transaccionales definidas en la especificación 1.0.0 del *Web Feature Service*, siendo estas GetCapabilities, DescribeFeatureType, GetFeature y las operaciones con filtros lógicos simples, aritméticos, de comparación (a excepción de NullCheck) y un conjunto reducido de los operadores espaciales (Dwithin, Intersects y BBOX), mientras que Geoserver, además de implementar todas las operaciones no transaccionales, también implementa las operaciones transaccionales GetFeatureWithLock, Query, Insert, Delete, Update y Lock. Por otro lado, Mapserver implementa las versiones 1.0.0, 1.1.0 y 1.1.1 de la especificación del *Web Map Service*, soportando las operaciones básicas GetCapabilities, GetMap y GetFeatureInfo; y las operaciones con SLDs

DescribeLayer, GetLegendGraphic, GetStyles (en la parte del servidor) y PutStyles (en la parte del cliente). Geoserver por su parte, en las versiones 1.0.0, 1.1.0 y 1.1.1, únicamente implementa las operaciones básicas.

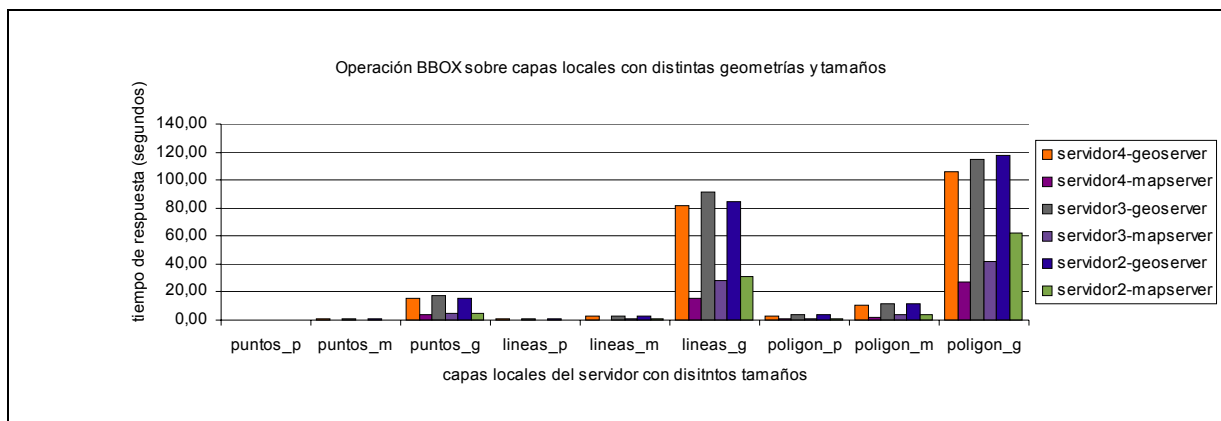


Figura 6: Comparativa de la operación espacial BBOX realizada sobre WFS con datos locales en los distintos nodos.

Como muestran las figuras 6 y 7, Mapserver ofrece tiempos de respuesta muy inferiores a los de Geoserver ante las peticiones realizadas a los *Web Feature Services* y *Web Map Services*. También se observa que para una misma solución, el componente hardware es un elemento clave en el tiempo de respuesta, siendo menos relevante el sistema operativo elegido. Por otro parte, en la figura 9 se puede observar como el incremento de usuarios pidiendo la misma capa de forma simultánea reduce considerablemente el número de peticiones por segundo que es capaz de responder el servicio. Además, también puede verse que la utilización de la base de datos como fuente de datos (figura 8) supone un impacto negativo en la eficiencia del servicio.

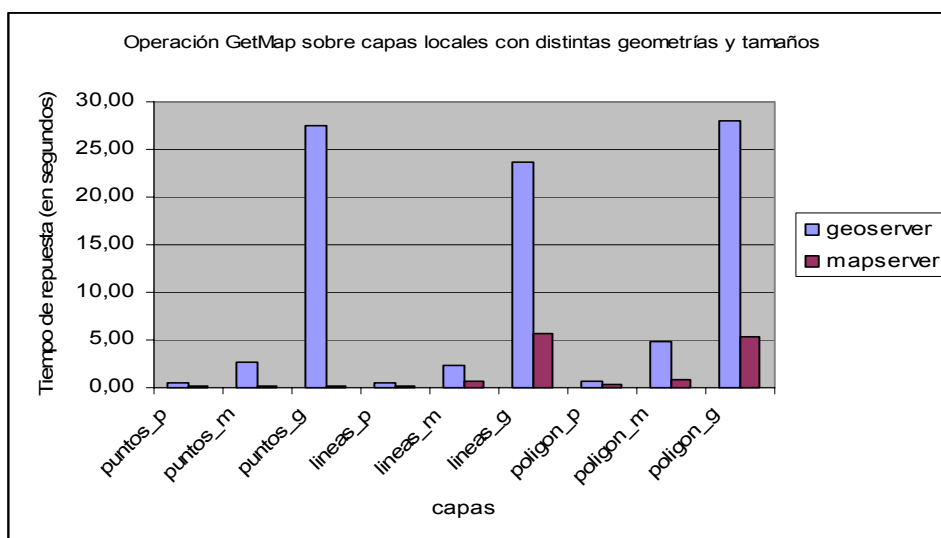


Figura 7: Comparativa de la operación GetMap realizada sobre WMS con datos locales en los distintos nodos.

En general, para los dos geo-servicios evaluados se ha observado que Mapserver ofrece un rendimiento muy superior al obtenido por Geoserver en todas las pruebas realizadas, independientemente del hardware, sistema operativo, fuente de datos o carga de usuarios utilizado. Esta superioridad de Mapserver es debida principalmente al hecho de que Mapserver haya sido programado con C++, lo que le permite un acceso más eficiente a los recursos hardware del sistema. El acceso simultáneo por parte de varios usuarios a una misma capa supone una pérdida de prestaciones, especialmente cuando el volumen de información es grande, llegándose a producir una situación la que el servidor ya no puede atender más peticiones. Esta situación degenera más rápidamente cuando se utiliza la base de datos debido al tiempo extra necesario para que el geo-servicio conecte con esta, se autentique, el gestor de la base de datos realice la consulta, se reciban los resultados provenientes de la base de datos y posteriormente sean transformados del formato nativo de PostGIS al formato interno de cada solución.

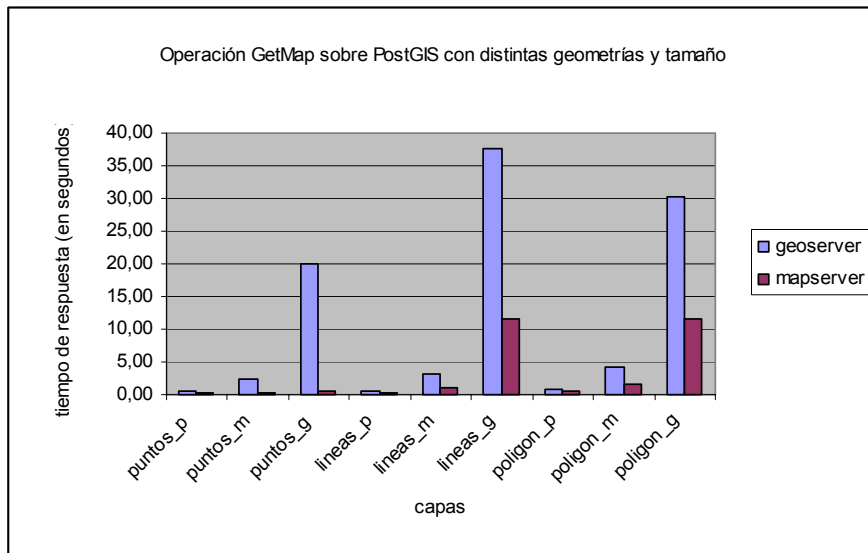


Figura 8: Comparativa de la operación GetMap realizada sobre WMS con datos de PostGIS.

También se ha observado en el caso de los *Web Feature Services* que una fracción importante del tiempo necesario para satisfacer las peticiones depende de la velocidad con la que los datos pueden ser transformados en GML. El ancho de banda de la red también juega un papel importante, por lo que la transmisión de varios cientos de megabytes a través de redes con conexiones lentas como el ADSL se convierte en el cuello de botella.

La especificación de implementación de los *Web Feature Services*, en su versión 1.0.0 y 1.1.1 [6] permite que se utilicen otros formatos de salida a parte de GML. Esto abre las puertas para que el WFS pueda enviar la respuesta en un formato comprimido. Los experimentos realizados con los servidores Linux y la herramienta de compresión *gzip* (simulando un único usuario) muestran que utilizando el nivel más bajo de compresión es posible reducir entre una quinta y una décima parte el mensaje original, siendo el tiempo necesario para su compresión varios órdenes de magnitud menor que el tiempo necesario para la transmisión del mensaje original. Sin embargo, la reducción del tiempo total de transferencia y ancho de banda consumidos se hace a costa de una mayor sobrecarga del servidor y una menor interoperabilidad, ya que el receptor ha de saber de antemano qué tipo de mensaje está recibiendo.

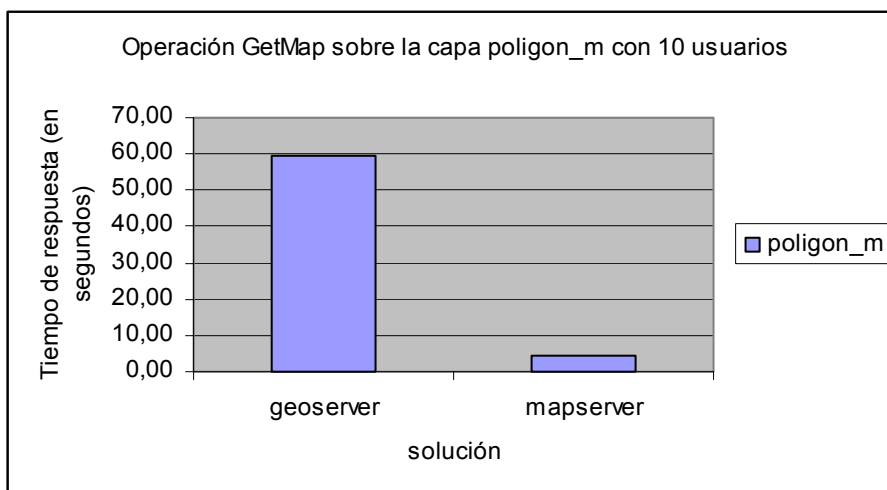


Figura 9: Comparativa de la operación GetMap realizada sobre WMS con datos locales y 10 usuarios simultaneos solicitando la capa poligon_m.

CONCLUSIONES

En este artículo hemos presentado los primeros resultados de un estudio que pretende valorar de forma objetiva el estado de madurez de las soluciones *open source* implementadas de los distintos componentes que conforman la arquitectura de las IDEs. Estas soluciones han sido probadas bajo una variedad de condiciones distintas como son el número de usuarios en el sistema, la carga de datos, el sistema operativo, las licencias de uso, portabilidad, tiempos de respuesta y lenguaje de programación utilizado para su implementación.

Los resultados preliminares obtenidos apuntan en términos generales a que tanto en el caso de los *Web Feature Services* como de los *Web Map Services*, Mapserver es siempre más eficiente que Geoserver, tanto en el uso de datos almacenados localmente como de la base de datos espacial, viéndose con mayor claridad cuanto mayor es la carga de usuarios o el volumen de información solicitada. También se ha visto en ambos casos (WMS y WFS), que la utilización de la base de datos supone para todas las soluciones una fuerte penalización en el tiempo total de respuesta y que el sistema operativo elegido no supone una gran diferencia en los resultados obtenidos. En el aspecto cualitativo, Mapserver es la solución que implementa la mayor parte de las especificaciones de los *Web Map Services*, mientras que es Geoserver el que en el caso de los *Web Feature Services* el que implementa en su totalidad las especificaciones.

El trabajo donde se enmarca este artículo fue concebido pensando en un conjunto reducido de variables a probar. Sin embargo, el número de permutaciones posibles que se pueden dar al combinar los distintos valores de estas variables (ej. incrementar la cantidad de memoria RAM en el servidor, considerar otras soluciones existentes o utilizar distintas cargas de usuarios) han hecho de esta tarea un proceso arduo de llevar a cabo.

En un futuro se incluirán más comparativas referentes a los geo-servicios tratados en este artículo, como el uso de las reproyecciones, de distintos formatos raster, de SLDs o el consumo de memoria por parte de los geo-servicios; así como la inclusión de otras soluciones no contempladas como Deegree (<http://deegree.sourceforge.net>); y comparativas con otras soluciones existentes para el resto de los componentes que conforman la arquitectura de las IDEs (clientes pesados, servicios de catálogos y repositorios de información geoespacial).

Por último, en un plazo relativamente corto, se pretende publicar un informe que abarque de forma más extensa los resultados aquí presentados. Esta será una contribución más a una IDEE abierta y heterogénea respecto a su implementación, sin que este estudio pretenda menoscabar las soluciones propietarias ya existentes en el mercado.

REFERENCIAS

1. Harrison, J., (2002): "OGC Web Services - Geoprocessing and the New Web Computing Paradigm". Geoinformatics, 5: 18-21
2. INSPIRE Initiative. <http://inspire.jrc.it> (último acceso: Noviembre 2005)
3. Smits, Paul et al (2002): INSPIRE Architecture and Standards Position Paper. Architecture and Standards Working Group. http://inspire.jrc.it/documents/inspire_ast_pp_v4_3_en.pdf (último acceso: Noviembre 2005).
4. OCCAMLAB. <http://www.occamlab.com/conan/index.jsp> (último acceso: Junio 2004).
5. Proyecto ACE-GIS. <http://www.acegis.net> (último acceso: Junio 2005).
6. Vretanos, Panagiotis A. (2002): OpenGIS® Web Feature Service (WFS) Implementation Specification. https://portal.opengeospatial.org/files/?artifact_id=8339 (último acceso: Noviembre 2005)
7. Whiteside, Arliss (2005): OpenGIS® Web Service Common Implementation Specification. https://portal.opengeospatial.org/files/?artifact_id=8798 (último acceso: Noviembre 2005).
8. Geospatial Interoperability Reference Model (GIRM). <http://gai.fgdc.gov/girm> (último acceso: Noviembre 2005)
9. Federal Geographic Data Comité (FGDC): <http://www.fgdc.gov> (último acceso: Noviembre 2005)
10. Alameh, Nadine (2003): "Chaining Geographic Information Web Services," IEEE Internet Computing, vol. 07, no. 5, pp. 22-29.
11. de La Beaujardiere, Jeff (2004): OpenGIS® Web Map Service (WMS) Implementation Specification. http://portal.opengeospatial.org/files/?artifact_id=5316 (último acceso: Noviembre 2005).