

# Medición de rendimientos de servicios WMS con JMeter

Deepak P. Daswani<sup>1</sup>, J. J. Rodrigo<sup>1</sup> y J. Rosales<sup>2</sup>

<sup>1</sup>Depto. de Ingeniería  
GRAFCAN. Cartográfica de Canarias, S.A  
C/ Panamá 34, Naves 8 y 9  
Santa Cruz de Tenerife  
{ddaswani, jrodrigo}@grafcan.com

<sup>2</sup>Dtor. Técnico  
GRAFCAN. Cartográfica de Canarias, S.A  
C/ Panamá 34, Naves 8 y 9  
Santa Cruz de Tenerife  
jrosales@grafcan.com

## Resumen

En este documento se describe de forma detallada el proceso de análisis y medición de rendimientos de los distintos servicios WMS mediante la herramienta JMeter, que permite testear cada uno de los diferentes parámetros de configuración de la infraestructura utilizada para albergar dichos servicios y determinar los valores óptimos hasta dar con la arquitectura final soportada.

**Palabras clave:** JMeter, WMS, rendimiento, análisis, arquitectura

## 1 Introducción

A la hora de planificar el desarrollo de un sistema de información, uno de los principales puntos a resolver y proceso clave para garantizar el éxito del mismo es el de dimensionar correctamente la estructura a implantar, tanto en lo que a hardware como a software se refiere, para poder atender correctamente el número de usuarios esperado y proporcionar un servicio adecuado con la calidad esperada.

Es necesario poder someter al sistema a una amplia batería de pruebas exhaustiva que permita obtener medidas concretas del rendimiento del mismo, del número de usuarios que es capaz de soportar, los tiempos de respuesta que es capaz de ofrecer en diferentes situaciones de carga, así como diagnosticar el comportamiento que va a tener en un entorno de producción.

Si hablamos de un sistema de información web en Internet, basado en el estándar HTTP (como el utilizado por un servicio WMS), es aún mayor la necesidad de contrastar debidamente la capacidad soportada por dicho sistema y conocer el límite de funcionamiento que posee, ya que una planificación incorrecta de los recursos que intervienen en la puesta en marcha del mismo dar lugar a unos resultados nefastos en situaciones de estrés aunque el código fuente de los aplicativos que corren en él pueda estar optimizado al máximo .

Para poder despejar esta incógnita que se plantea, existen aplicaciones que se encargan de simular situaciones de carga de usuarios sobre cualquier sistema. Dichas aplicaciones se encargan de realizar peticiones a un sistema y medir todos los factores que intervienen en la respuesta del mismo para proporcionar medidas de tiempo, carga, rendimiento... Además estas aplicaciones tienen un alto grado de flexibilidad y permiten simular situaciones reales de carga parametrizando prácticamente todos los factores que intervienen en una navegación real.

Una de estas aplicaciones de medición, que se encuentra entre las más famosas en su categoría y además es de carácter gratuito es JMeter [1], que es la que se ha utilizado para realizar la minuciosa y compleja batería de pruebas sobre IDECanarias [2], permitiendo analizar detalladamente el comportamiento del entorno montado ante diferentes escenarios, y comprobar la repercusión que la alteración de diversos parámetros tienen en el rendimiento del mismo, para ir depurando y eligiendo los mejores valores para los mismos hasta llegar a la arquitectura final.

## **2 Descripción y funcionamiento de JMeter**

JMeter es una aplicación del proyecto Apache Jakarta [3] escrita en Java, y permite como hemos comentado anteriormente realizar pruebas de rendimiento y de funcionalidad sobre aplicaciones tipo cliente/servidor escritas en cualquier lenguaje. Puede ser utilizado para realizar pruebas sobre conexiones de bases de datos JDBC, FTP, LDAP, Servicios WEB, HTTP, y conexiones TCP genéricas.

Como ya sabemos, el WMS [4] es un estándar que utiliza el protocolo HTTP [5], y la navegación por servicios WMS a través de un visor es traducida en peticiones HTTP que pueden ser trasladadas a cualquier navegador. Por esto, las pruebas sobre IDECanarias se han realizado simulando peticiones HTTP sobre los distintos servidores modificando diferentes parámetros.

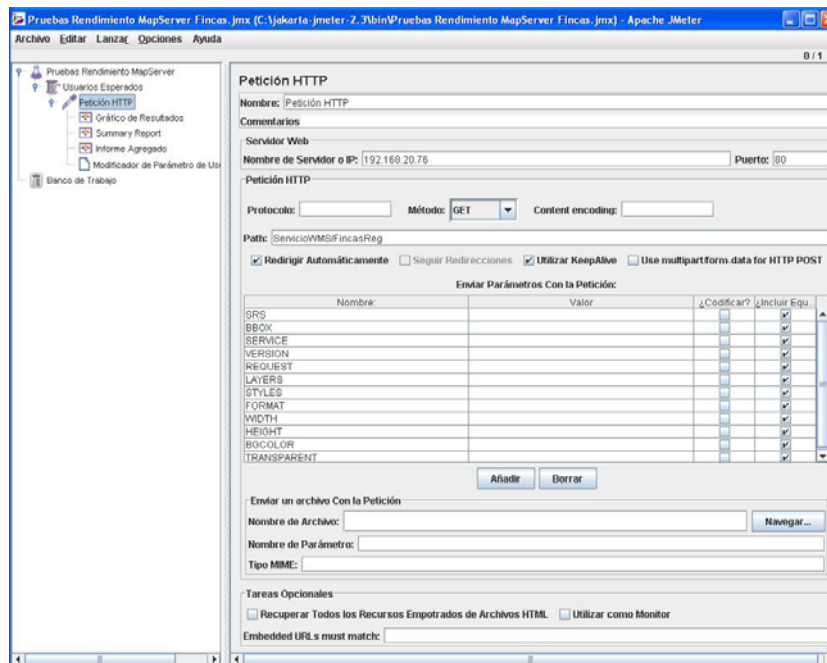


Figura 1. Aspecto visual de JMeter a la hora de analizar una petición HTTP

JMeter permite establecer como variables principales a la hora de realizar una simulación:

- *Número de usuarios:* Se corresponde con el número de procesos concurrentes que se van a lanzar.
- *Tiempo de subida:* Es el tiempo en segundos que tardan en iniciarse todos los procesos. Si el tiempo es cero, se lanzan todos a la vez. Este parámetro es importante.

- *Número de iteraciones*: Es el número de veces que se van a lanzar los n usuarios determinados en el primer parámetro. Si es mayor que uno, cada iteración utilizará el parámetro tiempo de subida para lanzar los siguientes n usuarios.

La modificación y alteración correcta de estas tres variables permite simular diversas tipologías y situaciones de carga de los servidores.

A la hora de simular peticiones HTTP, se debe especificar la dirección del servidor o *host* al que se va enviar la petición y la URL a la que se quiere realizar la petición. Pueden incluirse los parámetros a enviar por *querystring* en la propia cadena de la URL, o se pueden especificar y determinar el método de envío POST, para que simule peticiones hechas desde un formulario.

Una característica que hace a JMeter realmente potente es la posibilidad de especificar un fichero de usuarios en formato XML (llamado *users.xml*) que permite establecer parámetros variables para las peticiones, en aras de simular un escenario real en el que todas las peticiones que recibe el servidor no son las mismas y evita así posibles situaciones de cacheo. De esta forma si se añaden al fichero *users.xml* un número n de *threads* con conjuntos de diferentes parámetros, y se especifica a la hora de la ejecución un número mayor de usuarios o de iteraciones, las peticiones irán alternándose con esos n usuarios de manera rotativa, lo que permite simulaciones más reales al variar continuamente los parámetros pasados en la petición HTTP al servidor.

JMeter permite infinidad de opciones para manejar los resultados obtenidos de procesar la petición HTTP. En este artículo veremos las más importantes y las que hemos utilizado para las pruebas sobre IDECanarias.

## 2.1 Gráfico de Resultados

Muestra un gráfico con los resultados obtenidos de ejecutar las peticiones HTTP especificadas ofreciendo en una gráfica de líneas los valores de tiempo en milisegundos, rendimiento, y los valores de media, mediana y desviación típica.

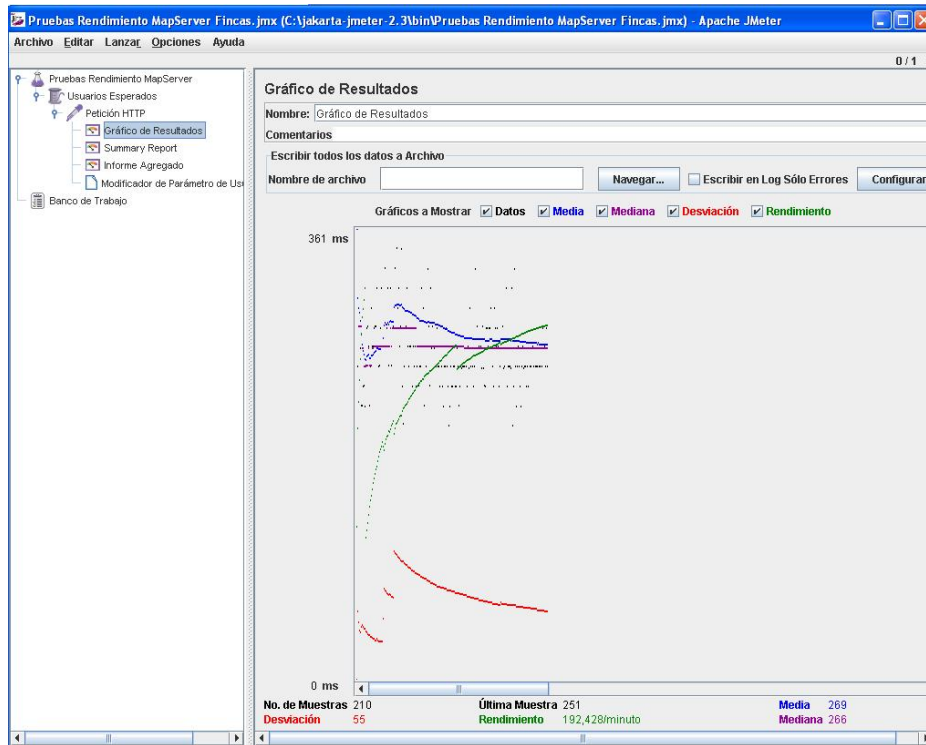


Figura 2. Gráfico de Resultados tras realizar una petición HTTP

## 2.2 Summary Report

Es el punto donde se presenta la tabla con la información que resulta de las peticiones HTTP, y que permite interpretar los resultados obtenidos y establecer comparaciones entre pruebas. En esta tabla se muestran los siguientes datos:

- Número de muestras
- Tiempo medio en milisegundos
- Tiempo mínimo en milisegundos
- Tiempo máximo en milisegundos
- Desviación típica
- Porcentaje de Error
- Rendimiento

- Kb/segundo
- Media de Bytes recibidos

Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
Petición HTTP	10	241	218	328	29,13	0,00%	4,1/sec	302,91	75994,0
TOTAL	10	241	218	328	29,13	0,00%	4,1/sec	302,91	75994,0

Figura 3. Datos obtenidos por el Summary Report tras una simulación con JMeter

Estos valores son los que permiten analizar los resultados obtenidos estableciendo comparaciones entre numerosas ejecuciones variando los parámetros principales de número de usuarios, tiempo de subida e iteraciones y observando cómo oscilan los tiempos mínimos, medio y máximo resultantes, así como la medida de rendimiento que JMeter ofrece.

### 2.3 Informe agregado

Presenta la tabla anterior pero mostrando los resultados agregados. En este caso los valores que se muestran son:

- Número de muestras
- Tiempo Medio en segundos
- Mediana
- Línea de 90%
- Tiempo mínimo en segundos
- Tiempo máximo en segundos
- Rendimiento
- Total de kilobytes recibidos

### 2.4 Guardar respuestas en un archivo

Esta opción permite guardar los resultados de todas las peticiones HTTP realizadas en archivos de salida. Se trata de un mecanismo útil para depurar posibles errores y asegurarnos que las peticiones están teniendo el resultado esperado.

## 2.5 Árbol de Resultados

Este punto permite visualizar con detalle cada petición HTTP realizada analizando las cabeceras de la petición y de la respuesta obtenidas. De este modo, podemos monitorizar qué sucede a la hora de procesar cada petición.

Como hemos comentado anteriormente, existen multitud de opciones más para procesar los resultados obtenidos (disponibles en JMeter's Wiki [6]), pero las descritas anteriormente han sido las más relevantes para el conjunto de pruebas realizado sobre IDECanarias.

## 3 Dinámica de Pruebas

El procedimiento aplicado para monitorizar el rendimiento de las diferentes configuraciones a testear sobre IDECanarias ha consistido en tomar los datos del Informe de resultados (Summary Report) obtenidos a partir de repetidas ejecuciones de peticiones WMS, escenificando para cada conjunto de parámetros diferentes situaciones de carga, incrementando progresivamente el número de usuarios, así como el tiempo de subida para los mismos o el número de iteraciones. De este modo se obtienen diferentes valores de rendimiento, tiempo medio, máximo y mínimo de ejecución, así como tasa de transferencia en kilobytes/seg, número de Kbs recibidos u otros valores que nos permiten establecer una pauta de comportamiento y analizar dichos resultados desde una perspectiva gráfica.

Los parámetros que se han ido alterando para poder reproducir un variado abanico de peticiones y conocer la repercusión que éstas tienen en el resultado devuelto consisten en la variación de la capa mostrada, el sistema de referencia así como el formato o el tamaño de la imagen. Asimismo, para cada una de estas diferentes condiciones, se han establecido seis diferentes tipos de "usuarios", representados por diferentes *threads* con diferentes valores de Bounding Box.

Este conjunto de pruebas se han repetido sobre diferentes configuraciones de sistema para poder comparar aquellos factores que intervienen en el rendimiento del mismo y poder ir refinando el modelo inicial hasta poder dar con la configuración óptima para cada recurso.

Un ejemplo de resultados obtenidos tras repetir varias ejecuciones para decidir entre diferentes configuraciones se muestra en las siguientes figuras:

### Configuración 1

Nº Procesos	T.Subida	Iterac.	P.Finales	Med	Min	Max	Error %	Rdto.	Kb/Sec	Avg Bytes
1	0	50	50	428	10 9	1110	0	2,33	83,20	36614,3 8
2	0	40	80	482	10 9	1204	0	4,07	146,2 9	36811,9 3
4	0	20	80	680	11 0	2407	0	5,52	198,5 3	36811,9 3
8	0	10	80	1243	94	7219	0	5,39	193,7 3	36811,9 3
16	0	10	160	2249	14 0	10376	0	5,71	206,5 2	37011,5 7
24	0	10	240	3590	12 5	16643	0	5,51	199,7 3	37141,1 7
32	0	10	320	4555	12 5	22800	0	5,64	204,2 1	37058,8 6
40	0	8	320	5630	12 5	29394	0	5,69	205,9 7	37058,8 6
50	0	6	300	6606	14 0	31879	0	6,00	217,8 0	37141,1 7
64	0	4	256	7477	12 5	33176	0	5,99	216,8 6	37060,1 7
128	1	4	512	1526 3	12 5	70383	0	5,92	214,2 9	37089,7 2
256	1	4	1024	3092 1	12 5	13104 8	0	5,81	210,7 0	37120,9 2

Figura 4. Tabla de resultados obtenidos tras varias simulaciones con la Configuración .1



### Configuración 2

Nº Procesos	T.Subida	Iterac.	P.Finales	Med	Min	Max	Error %	Rdto.	Kb/Sec	Avg Bytes
1	0	50	50	348	94	844	0	2,86	102,35	36614,38
2	0	40	80	414	93	1796	0	4,22	151,63	36811,93
4	0	20	80	361	93	922	0	9,77	351,28	36811,93
8	0	10	80	624	93	5546	0	9,47	340,27	36811,93
16	0	10	160	1179	93	9015	0	9,84	355,57	37011,57
24	0	10	240	1769	93	13124	0	10,58	383,72	37141,17
32	0	10	320	2424	93	15202	0	10,76	389,51	37058,86
40	0	8	400	2990	93	22342	0	10,24	370,77	37089,33
50	0	6	300	3612	93	21014	0	10,71	388,42	37141,17
64	0	4	256	4511	93	19967	0	10,14	366,95	37060,17
128	1	4	512	8864	93	42388	0	10,33	374,08	37089,72
256	1	4	1024	17589	93	91713	0	10,39	376,77	37120,92

Figura 5. Tabla de resultados obtenidos tras varias simulaciones con la Configuración.2

### Configuración 3

Nº Procesos	T.Subida	Iterac.	P.Finales	Med	Min	Max	Error %	Rdto.	Kb/Sec	Avg Bytes
-------------	----------	---------	-----------	-----	-----	-----	---------	-------	--------	-----------

1	0	50	50	404	94	953	0	2,47	88,28	36614,38
2	0	40	80	492	125	1250	0	3,98	143,11	36811,93
4	0	20	80	659	110	1735	0	5,81	208,90	36811,93
8	0	10	80	1221	109	3969	0	5,81	208,90	36811,93
16	0	10	160	2359	110	10345	0	5,49	198,32	37011,57
24	0	10	240	3646	141	18190	0	5,62	203,90	37141,17
32	0	10	320	4638	110	21440	0	5,88	212,71	37058,86
40	0	8	320	5616	94	27613	0	5,74	207,65	37058,86
50	0	6	300	6556	125	28223	0	5,90	213,92	37141,17
64	0	4	256	7978	125	34379	0	5,65	204,45	37060,17
128	1	4	512	15258	125	67039	0	5,94	215,22	37089,72
256	1	4	1024	31881	125	157441	0	5,63	204,16	37120,92

Figura 6. Tabla de resultados obtenidos tras varias simulaciones con la Configuración.3

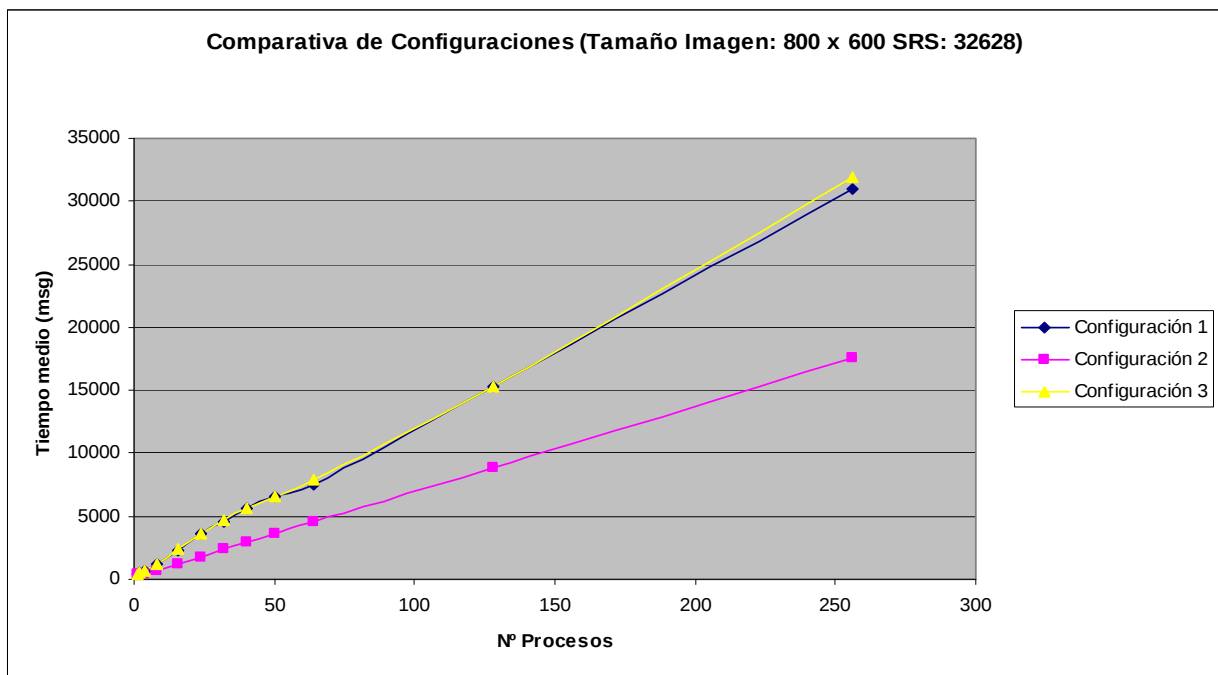


Figura 7. Gráfica resultante de comparar las tablas de las simulaciones de las distintas configuraciones

En este ejemplo se aprecia claramente que la configuración 2 es la que mejor se ajusta para nuestro propósito de minimizar el tiempo medio de respuesta. En este caso concreto observamos en las tablas que tanto los tiempos máximos son inferiores dicha configuración, pero podría darse el caso de que esto no fuera así y por lo tanto tampoco fuera la seleccionada para implantar en la solución final.

Esta dinámica de pruebas ha permitido partir de una situación inicial y apreciar qué repercusiones tiene cada optimización o cambio introducido en el sistema para poder ir descubriendo las ventajas e inconvenientes del mismo, y analizarlo desde diferentes puntos de vista estableciendo la relación entre diferentes parámetros WMS, parámetros de ejecución (número de usuarios, tiempo de subida, número de iteraciones) y configuraciones de entorno (servidores, versión de software, ubicación de los datos...).

La realización de esta amplia batería de pruebas utilizando JMeter ha permitido estudiar y analizar minuciosamente cada detalle de la infraestructura de IDECanarias. Por un lado, se han evaluado numerosos factores que influyen en la configuración del hardware, como cambios de estructura de servidores físicos, servidores virtuales, número de servidores, estructuras de almacenamiento de los datos, disposición de los mismos, accesos desde diferentes topologías de red... Por otro lado, se han introducido cambios desde el punto de vista del software que afectan a los ficheros de configuración de mapas, optimizaciones en el código fuente de MapServer [7], cambios de parámetros de los servidores web.

Todos estos factores han sido testeados de manera concreta variando diferentes parámetros y escogiendo la configuración óptima para el mismo, permitiendo relacionar cada paso con el anterior hasta adoptar una arquitectura final para IDECanarias, basando cada decisión tecnológica en un fundamento sustentado por un exhaustivo proceso de pruebas, que por otra parte se halla documentado, permitiendo su posible utilización a la hora de introducir cambios u optimizaciones en el futuro.

## Referencias

[1] Apache JMeter <http://jakarta.apache.org/jmeter/>

[2] IDECanarias, <http://www.idecan.grafcan.es/idecan>

- [3] The Apache Jakarta Project, <http://jakarta.apache.org/>
- [4] Web Map Service (WMS), <http://www.opengeospatial.org/standards/wms>
- [5] HTTP Protocol, <http://www.w3.org/Protocols/>
- [6] JMeter Jakarta Wiki, <http://wiki.apache.org/jakarta-jmeter/>
- [7] Mapserver Project homepage, <http://mapserver.gis.umn.edu/>