

MOBILE WEB SERVICES PERFORMANCE ANALYSIS**ANÁLISIS DE RENDIMIENTO EN SERVICIOS WEB MÓVILES****Eduardo Carrillo Zambrano, Jesús E. Ortega Arévalo, Guillermo Sánchez Jaimes****Universidad Autónoma de Bucaramanga**Facultad de Ingenierías
jeortegaa@hotmail.com

Abstract: The aim of the present study is to generate a few tests tending to measure the use of resources in the ends, principally in the server, trying to optimize the use of such a scanty bandwidth in the mobile devices. For such purpose there use two algorithms of standard compression in the platforms Microsoft Windows and in Linux, the gzip and the XMLPPM. It generates recommendations and give concepts about current options to implement Web Services. The Introduction contains information about the importance of the study; it continues with the State of the art, which presents a vision of what already exists in evaluation of the yield and conceptualization in Mobile Web Services; It continues with the Development of the work that clarifies the goal in the study, the methodologies of work and the schemes of test bench; it contains then a chapter of conclusions that formalizes a series of definitions and recommendations seeking to extract the major profit of the results of the tests and with the intention of generating recommendations to optimize the use of the bandwidth in mobile devices; the document finishes with the Bibliography, a list of documentary references, theoretical support to the study.

Resumen: El objetivo del presente estudio es generar unas pruebas tendientes a medir el uso de recursos en los extremos, principalmente en el servidor, cuando se pretende optimizar el ahorro en el consumo del ancho de banda tan escaso en los dispositivos móviles. Para tal fin se emplean dos algoritmos de compresión estándar en las plataformas Microsoft Windows y Linux, el gzip y el XMLPPM. Genera recomendaciones y ambienta con respecto a las actuales opciones para implementar Servicios Web. Contiene una Introducción, que define la importancia del estudio; sigue con el Estado del arte, que presenta una visión de lo que ya existe en evaluación del rendimiento y compara objetivos de los diferentes estudios; sigue con el Desarrollo de la propuesta que aclara a qué se quiere llegar en el estudio, las metodologías de trabajo y los esquemas de banco de pruebas; contiene luego un capítulo de conclusiones que formaliza una serie de definiciones y recomendaciones buscando sacar el mayor provecho de los resultados de las pruebas y con el ánimo de generar sugerencias para optimizar el aprovechamiento del ancho de banda en dispositivos móviles; el documento termina con las referencias, una lista documental de apoyo teórico al estudio.

Keywords: Web services, Mobile web, Service performance analysis.

1. INTRODUCCIÓN

Los Servicios Web son una tendencia que busca automatizar la comunicación entre los dispositivos computacionales que usan la pila de protocolos TCP/IP para emitir servicios o recibirlos.

Existen ya opciones en muchas plataformas para emitir este tipo de servicios y vemos en Microsoft Windows .NET Framework y en sistemas tipo UNIX el proyecto Apache Axis.

Ahora, existe paralelamente una creciente expectativa y uso hacia dispositivos que permiten implementar los conceptos de movilidad. Este concepto se ve representado principalmente en los Personal Digital Assistant (PDAs) de los cuales los más representativos son los Palm de la empresa Palm y los Pocket PC que provienen de varios fabricantes pero que se caracterizan por tener como sistema operativo el Windows CE.

Habida cuenta del gran auge de la tecnología de Servicios Web y de la importancia que reviste el poder automatizar la comunicación en Internet o, viendo otras de sus características más importantes, la posibilidad de solicitar o recibir sólo lo que realmente se requiere en un momento dado, y del crecimiento en la importancia, uso, disponibilidad de recursos y desarrollos en el mundo de los PDAs, se hace muy necesario realizar un análisis del rendimiento de los Servicios Web, tendiente a optimizar el aprovechamiento del ancho de banda de los canales a los cuales se conectan los PDAs. Este análisis de rendimiento busca generar algunas recomendaciones con respecto a la forma en que se puede aprovechar mejor el ancho de banda, dar una idea clara de las tecnologías disponibles en el lado del servidor y en el lado de los clientes, y concluir el costo/beneficio de usar las técnicas recomendadas en el aprovechamiento óptimo del consumo del ancho de banda en los canales móviles.

2. ESTADO DEL ARTE

Debido a que la tendencia a Servicios Web Móviles es relativamente nueva, y que surgió ante las inmensas posibilidades que se observaron en los Servicios Web al permitir que dispositivos con escasos recursos computacionales y de conectividad accedieran a información bastante específica por medio de un protocolo muy liviano como el HTTP, realmente no existen estudios enfocados hacia un análisis del rendimiento con el objetivo de aprovechar de manera óptima los escasos recursos de conectividad para los dispositivos móviles representados principalmente

por los Personal Digital Assistant (PDA). Existe un primer artículo [TIA01], que egenera un análisis del uso de la conectividad en ambientes móviles, específicamente para PDAs, aunque no realiza un análisis del consumo en el procesamiento en los extremos. Este estudio que data del año 2003, no estaría muy actualizado en cuanto a las características que se pueden encontrar actualmente en el ambiente móvil o en la arquitectura de servidores. Adicionalmente realiza pruebas en una sola Plataforma, Microsoft IIS.

Un segundo trabajo [BAN01], que usa dos ejemplos de clientes de Servicios Web elaborados en Java, elabora unas pruebas centradas en definir el uso que estos Servicios Web hacen del ancho de banda. No genera análisis de consumo de procesamiento ni ningún método de optimización del uso del ancho de banda.

Finalmente, un tercer estudio [WON01], basa todo su recorrido durante las pruebas en la creación de Servicios Web en dispositivos móviles usando J2ME, la plataforma Java para dispositivos con escasos recursos como el caso de los teléfonos móviles con un ambiente de programación, los PDAs, y los smartphones que son una combinación de teléfonos móviles y PDAs.

El presente estudio es genera pruebas tendientes a medir el uso de recursos en los extremos, principalmente en el servidor, cuando se pretende optimizar el uso del ancho de banda tan escaso en los dispositivos móviles.

3. DESARROLLO DE LA PROPUESTA

3.1 Metodología

Se realizan investigaciones para documentación teórica en Bibliografía acerca de Servicios Web Móviles. Se hacen estudios acerca de los ambientes de desarrollo. Análisis de la composición de las solicitudes/respuestas con respecto a los protocolos SOAP y HTTP 1.1. Análisis del tamaño de las solicitudes/respuesta de dos ejemplos implementados. Análisis de la eficiencia en la compresión para los algoritmos gzip y XMLPPM. Creación de script que generen procesos paralelos de compresión en los ambientes Microsoft Windows y Linux. Creación de archivos tipo HTTP 1.1 y XML de tamaños en diferentes rangos. Análisis del uso de procesamiento en los extremos. Análisis de los resultados. Generación de recomendaciones.

3.2 Esquema de trabajo en los servicios WEB móviles

Los Servicios Web Móviles involucran los dispositivos móviles que tienen capacidad para conectarse a la red Internet y de poder implementar aplicaciones para consumir Servicios Web. La propia definición de Servicios Web [W3C01] asegura que el estándar se mantenga y que implementando clientes que sigan las recomendaciones bajo el protocolo SOAP 1.2, se puedan consumir los Servicios Web usando el Internet. Por tal motivo, la visión inicial de los Servicios Web como tradicionalmente se vería y se ilustra en la siguiente gráfica:

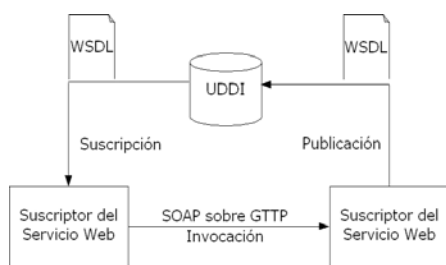


Fig. 1 Los estándares de Servicios Web en acción

Se convierten en un esquema móvil representado principalmente por PDAs como a continuación:

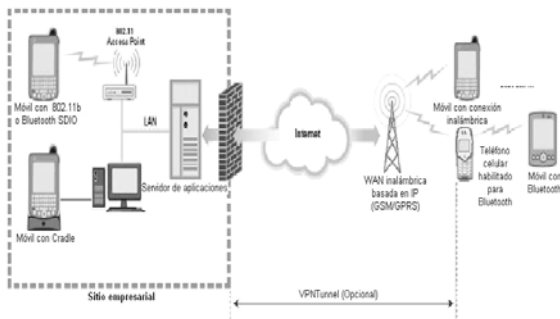


Fig. 2. Arquitectura de conectividad general de los Servicios Web Móviles⁹

La gráfica muestra de manera definitiva que la tecnología GSM/GPRS [GSM01] tiende a ser el estándar en la WWAN y que sin importar el dispositivo, al final se van a transportar paquetes. Existen adicionalmente varias organizaciones, además de la W3C, que buscan universalizar el ambiente de la movilidad, no sólo los Servicios Web y para referencias completas de cada una de las siguientes, incluyendo W3C, se puede visitar el sitio referenciado en la bibliografía de este artículo:

- OMA (Open Mobile Alliance) [OMA01]

- W3C (World Wide Web Consortium)[W3C02]
- WS-I (Web Services Interoperability Organization)[WSI01]
- OASIS (Organization for the Advancement of Structured Information Standards)[OAS01]

3.3 Banco de pruebas

El banco de pruebas comprende todos los elementos involucrados en el desarrollo de las pruebas. El siguiente gráfico y los párrafos que lo siguen lo describe con buen detalle:

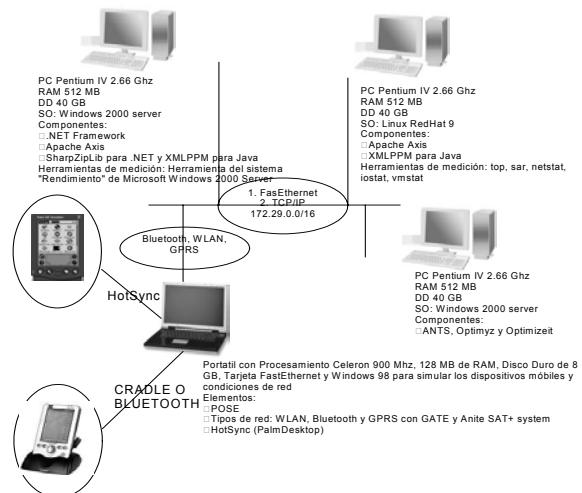


Fig. 3. Vista gráfica del Banco de Pruebas

3.3.1 Detalles del banco de pruebas

3.3.1.1 Hardware y Software de servicios

3.3.1.1.1 Publicación de Servicios Web en Windows

Descripción: Generación de Servicios Web en el ambiente Windows para ser consumidos por PDAs en el banco de pruebas; Arquitectura: PC x86; Procesamiento: Pentium IV 2.66 Ghz; Memoria RAM: 512 MB; Almacenamiento interno: IDE 40 GB; Sistema operativo: Microsoft Windows 2000 server con Service Pack 4; Antivirus: Norton

Componentes:

.NET Framework [MIC01]: La IDE de Windows para desarrollo y publicación de servicios en la Web; Apache Axis [APC01]: El componente de la organización Apache para publicar Servicios Web; gzip [GAI01] para .NET y XMLPPM [PPM01] para C: Compresores para el XML y HTML producido en el extremo del servicio; Herramientas de medición: Herramientas de medición: Herramienta del sistema "Rendimiento" de Microsoft Windows 2000 Server

⁹Tomado de [PLM01]

3.3.1.1.2 *Publicación de Servicios Web en Linux*

Descripción: Generación de Servicios Web en el ambiente Linux (Un tipo UNIX) para ser consumidos por PDAs en el banco de pruebas; Arquitectura: PC x86; Procesamiento: Pentium IV 2.66 Ghz; Memoria RAM: 512 MB; Almacenamiento interno: IDE 40 GB; Sistema operativo: Microsoft Windows 2000 server con Service Pack 4; Antivirus: Norton

Componentes:

ANTS[RDG01], Optimyz y Optimizeit[BRL01]

3.3.1.1.3 *Generación de carga*

Descripción: Simulación de carga en el consumo de Servicios Web. Se hace desde el sistema operativo Windows hacia los dos publicadores de Servicios Web en los ambientes Windows y Linux.; Arquitectura: PC x86; Procesamiento: Pentium IV 2.66 Ghz; Memoria RAM: 512 MB; Almacenamiento interno: IDE 40 GB; Sistema operativo: Linux RedHat 9

Componentes:

Apache Axis: El componente de la organización Apache para publicar Servicios Web; XMLPPM para Java: Compresores para el XML y HTML producido en el extremo del servicio; Herramientas de medición: top, sar, netstat, iostat, vmstat

3.3.1.1.4 *Conectividad para los PDAs y simulación de ambientes de conectividad*

Descripción: Tres objetivos: ejecutar los Emuladores de PDAs tanto Pocket PC como Palm, ofrecer conectividad a los PDAs y simular ambientes de conectividad Bluetooth, WLAN y GPRS; Portatil con Procesamiento Celeron 900 Mhz, 128 MB de RAM, Disco Duro de 8 GB, Tarjeta FastEthernet y Windows 98 para simular los dispositivos móviles y condiciones de red; Arquitectura: Portátil x86; Procesamiento: Intel Celeron a 900 Mhz; Memoria RAM: 128 MB; Almacenamiento interno: IDE 8 GB; Sistema operativo: Windows 98

Componentes:

POSE y Emulador Pocket PC: Emuladores de PDAs; Tipos de red: WLAN, Bluetooth y GPRS con GATE y Anite SAT+ system; HotSync (PalmDesktop): Sincronizar la instalación de aplicaciones para Palm

3.3.1.2 *Elementos clientes*

POSE: El Emulador para PALM; Emulador Pocket PC; Equipo real Palm IIIc; Equipo real Pocket PC Modelo iPAQ.

3.3.1.3 *Elementos para el desarrollo*

3.3.1.3.1 *Equipo usado:*

Hardware: Procesador: 1.5 Ghz; Memoria RAM: 1.5 GB; Disco duro: 36 GB; Interfaz de red: FastEthernet. Software: Sistema Operativo Microsoft Windows 2000 Server Service Pack 4

3.3.1.3.2 *Herramientas para aplicaciones sobre Palm*

Palm OS Developer Suite (La suite de desarrollo para Palm); Cygwin (Simulador de ambiente Unix para poder usar PRC-Tools en Windows); PRC-Tools (compiladores, enlazadores y otros para GCC); PilRC para Win32 (generador de formularios para Palm); Palm OS Emulator (simuladores de dispositivos Palm de acuerdo al ROM que se use)

3.3.2 *Narración del detalle de pruebas*

El banco de pruebas descrito obtiene los resultados tabulados y detallados más adelante.

3.3.2.1 *Tamaño en bytes de la información enviada y recibida*

3.3.2.1.1 *Tamaño original de archivos cliente en HTTP 1.1*

Esta prueba inicialmente ofrece tablas de la cantidad de bytes que transmite y recibe cada uno de los servicios. Existen dos ambientes, Windows y Linux. En cada ambiente existen dos ejemplos, una calculadora y un cálculo de factorial. En el ambiente Windows el ejemplo de calculadora se llama MathService y se desarrolla en VB (Visual Basic), el segundo ejemplo que se denomina FactorialServiceC está desarrollado en C#. En el ambiente Linux se desarrollan en C los dos ejemplos. Se usan en Windows el block de notas y en Linux el editor estándar vi para algunas modificaciones. Los ejemplos son obtenidos o modificados de ejemplos estándar en Internet.

La tabla 1 resume las características de los ejemplos:

Ambiente	Servidor	Lenguaje de desarrollo para cliente Calculadora	Lenguaje de desarrollo para cliente Calculadora	Nombres de ejemplo de calculadora	Nombres del ejemplo factorial
Windows	.NET Framework	Visual Basic	C#	MathService	FactorialServiceC
Linux	Apache Axis	C	C	Calculator	Factorial

Tabla 1. Características de los ejemplos de Clientes de Servicios Web para PDAs

3.3.2.2 La información HTTP 1.1 comprimida

Luego de esto, y de acuerdo a la importancia definida para el recurso de ancho de banda como parámetro fundamental de las mediciones, se concluye que un procedimiento adecuado es comprimir la información a enviar.

3.3.2.3 El uso del procesamiento

Realmente, el uso de procesamiento para la compresión o descompresión de los ejemplos en los dos servidores Windows y Linux es tendiente a cero, por lo cual se decide pasar a otra etapa de pruebas.

Una etapa que permite determinar cuál es el uso del procesamiento para rangos de tamaños de archivos y rangos de cantidad de procesos paralelos de compresión.

Es decir, se toman como base archivos de 1K, 10K, 100K y 1000K. Ya pensar en un archivo generado en XML de 1000K es una exageración, pues según lo investigado y probado, como máximo un archivo XML para un Servicio Web podría llegar a 10K. Esto se explica precisamente por una de las características más importantes y llamativas de los Servicios Web, el envío y recepción de sólo la información que se requiere. Para cada uno de estos archivos entonces se tabulan los usos de procesamiento para 1, 10 y 100 procesos en paralelo en el servidor.

3.4 Resultados de las pruebas

Las siguientes gráficas y párrafos resumen y explican los resultados obtenidos durante las pruebas:

Ambiente	Servidor	Lenguaje de desarrollo para cliente Calculadora	Lenguaje de desarrollo para cliente Calculadora	Nombres de ejemplo de calculadora	Nombres del ejemplo factorial
Windows	.NET Framework	Visual Basic	C#	MathService	FactorialServiceC
Linux	Apache Axis	C	C	Calculator	Factorial

Tabla 2. Características de los ejemplos de Clientes de Servicios Web para PDAs

La tabla 3 presenta el tamaño en bytes para el ejemplo MathServices

Tipo de mensaje	Mult.	Div.	Rest.	Suma
Solicitud SOAP	513	507	513	572
Respuesta SOAP	441	443	441	347

Tabla 3. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web MathService

La tabla 4 presenta el tamaño de lo emitido por el ejemplo FactorialServiceC:

Tipo de mensaje	Tamaño en bytes
Solicitud SOAP	502
Respuesta SOAP	414

Tabla 4. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web

FactorialServiceC

La siguiente tabla presenta el tamaño en bytes para el ejemplo Calculator.

Tipo de mensaje	Mult.	Div.	Resta	Suma
Solicitud SOAP	479	479	479	479
Respuesta SOAP	421	421	421	421

Tabla 5. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Calculator en Linux con Axis

La tabla 6 presenta el tamaño de lo emitido por el ejemplo Factorial:

Tipo de mensaje	Tamaño en bytes
Solicitud SOAP	476
Respuesta SOAP	443

Tabla 6. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Factorial en Linux con Axis

La tabla 7 resume las pruebas y los resultados obtenidos al ejecutar compresiones de archivos con formato XML en el servidor, con un sólo usuario concurrente.

Esto demuestra que para el ejemplo MathService, el consumo de procesamiento en el servidor, teniendo en cuenta que los archivos miden menos de 1K tiende a cero cuando hay una sola petición o respuesta.

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K	10580000	58541	180.73	Tiende a cero	Tiende a cero

Tabla 7. Resultados de comprimir XML en el servidor Windows con gzip con una sola petición

La tablas 8 y 9 muestran los resultados de las pruebas usando 10 y 100 procesos. En la tabla 8, se puede concluir que para 10 procesos concurrentes ejecutando una labor de compresión al máximo factor (gzip tiene factores de compresión del 1 al 9,

consumiendo más procesamiento el 9, pero generando mayor compresión), el consumo de procesamiento del algoritmo gzip tiende a cero sin importar qué tan grande es el archivo conteniendo HTTP 1.1 en la plataforma Microsoft Windows.

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K (10 MB)	10580000	58541	180.73	Tiende a cero	Tiende a cero

Tabla 8. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 10 peticiones

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Pico de procesamiento en compresión	Promedio en procesamiento para compresión
Archivo de 1K	1058	348	3.04	1.56	Tiende a cero
Archivo de 10K	10580	426	24.84	3.13	Tiende a cero
Archivo de 100K	105800	985	107.41	40.00%	5 segundos
Archivo de 1000K	1058000	6227	169.91	50.00%	10 segundos
Archivo de 10000K	10580000	58541	180.73	50.00%	60 segundos

Tabla 9. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 100 peticiones

En la tabla 9, que presenta un ambiente más acorde a lo que se presentaría en un ambiente real de producción de Servicios Web en Internet, se puede concluir que para 100 procesos concurrentes ejecutando una labor de compresión al máximo factor

(9 para gzip, consume más procesador), el consumo de procesamiento del algoritmo gzip tiende a cero en los casos de archivos del rango de 1Kbyte a 10Kbytes, lo normal en un ambiente de solicitud petición de Servicios Web transportando HTTP 1.1. Para

archivos mayores de 10Kbytes, en el rango de hasta 10000 Kbytes se denota un consumo de procesamiento que puede llegar a tener el procesador en un 50% en promedio durante un minuto. Las pruebas determinan que el procesamiento no estaría por encima del 85%, umbral que se considera ya peligroso para el tema de QoS, lo que percibiría el usuario. Ahora, 100 procesos concurrentes

ejecutando una labor de compresión con el algoritmo gzip sobre un archivo de 100Kbytes con contenido HTTP 1.1, consumirían el procesador en un 40% durante 5 segundos. Teniendo en cuenta que ya un archivo mayor de 10Kbytes es exagerado en Servicios Web, se puede concluir que el procesamiento no sería consumido inapropiadamente en este caso.

Método (Operación de la calculadora)	Tiempo para conectarse en milisegundos	Tiempo para recibir el primer byte en milisegundos	Tiempo para recibir el último byte en milisegundos	Bytes recibidos	Cantidad de respuestas del método
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Add	4	5	109	666	106
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Subtract	0	0	68	686	106
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Multiply	0	0	154	686	105
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Divide	0	0	99	681	105

Tabla 10. Resultados de la prueba de carga con ANTS al Servicio Web MathService en Microsoft Windows

La tabla 11 resume las pruebas y los resultados obtenidos al ejecutar compresiones de archivos con

formato XML en el servidor, con un sólo usuario concurrente.

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K (10 MB)	10580000	58541	180.73	Tiende a cero	Tiende a cero

Tabla 11. Resultados de comprimir XML en el servidor con XMLPPM con una sólo petición en Linux Axis

Esto demuestra que para el ejemplo Calculator, el consumo de procesamiento en el servidor, teniendo en cuenta que los archivos miden menos de 1K tiende a cero cuando hay una sólo petición o respuesta.

Las tablas 12 y 13 muestran los resultados de las pruebas usando 10 y 100.

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Tiempo tomado para comprimir
Archivo de 1K	1058	348	3.04	8.00%	Tiende a cero
Archivo de 10K	10580	426	24.84	9.00%	Tiende a cero
Archivo de 100K	105800	985	107.41	20.00%	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	30.00%	2 segundos
Archivo de 10000K (10 MB)	10580000	58541	180.73	100.00%	10 segundos

Tabla 12. Resultados de comprimir XML en el servidor con XMLPPM con 10 peticiones en Linux Axis

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Tiempo tomado para comprimir
Archivo de 1K	1058	348	3.04	60.00%	1 segundo
Archivo de 10K	10580	426	24.84	80.00%	1 segundo
Archivo de 100K	105800	985	107.41	80.00%	3 segundos
Archivo de 1000K (1 MB)	1058000	6227	169.91	100.00%	13 segundos
Archivo de 10000K (10 MB)	10580000	58541	180.73	100.00%	130 segundos

Tabla 13. Resultados de comprimir XML en el servidor con XMLPPM con 100 peticiones en Linux Axis

Aunque 10 MB es una exageración en cuanto a peticiones y respuestas en Servicios Web se refiere, se observa que XMLPPM es una opción que consume mucho más procesamiento que gzip. El equipo en el cual se instaló Linux y se hicieron las pruebas con XMLPPM tiene un procesamiento de 2800 Mhz, es decir un procesamiento mucho mejor que el equipo en el cual se hicieron las pruebas para Windows.

La siguiente es una comparación entre los niveles de compresión de gzip y XMLPPM en el equipo Linux:

Archivo	Gzip tamaño en bytes	XMLPPM tamaño en bytes
1K	257	227
10K	324	294
100K	774	706
1000K	5107	4707
10000K	48432	44615

Tabla 14. Comparación entre gzip y XMLPPM en Linux con respecto a compresión

Luego de este análisis se denota que la diferencia de compresión frente al consumo de procesamiento de la máquina no es significativa, a pesar de que XMLPPM comprime un poco más, pero consume mucho más procesamiento por el análisis de sintaxis que por defecto trae la herramienta para XML.

Realmente el análisis de sintaxis debe recaer sobre otras etapas del desarrollo.

Ahora, la siguiente tabla compara consumo de procesador para 100 sesiones en un archivo de 10 MB:

Archivo	Consumo XMLPPM	Consumo gzip
10000K	100% durante 130 segundos	100% durante 71 segundos

4. CONCLUSIONES

Usando algoritmos de compresión estándar como gzip permite reducir el uso de ancho de banda para un dispositivo móvil en un 66% en el momento de acceder a un Servicio Web. El tema del procesamiento no es una preocupante en este caso según lo analizado pues el tamaño de los archivos que generalmente se generan en este ambiente de Servicios Web contienen tamaños que no exigen que los algoritmos de compresión requieran usos excesivos de procesamiento en los extremos.

Se determinó que el servidor es el que puede sufrir un poco más el uso de esta técnica, pero se concluyó luego de analizar el contenido de las peticiones y respuestas de un Servicio Web en dos ejemplos particulares, que lo que se transporta por la red (Internet) es contenido en formato HTTP 1.1, lo que permite usar los algoritmos de compresión estándar como el gzip para hacer que la información que viaje por el canal sea reducido.

El tamaño promedio de las solicitudes/peticiones de Servicios Web no sobrepasan los 10000 bytes, es decir, los tamaños de archivos que viajan por la red y que necesitan ser transportados en formato de protocolo HTTP 1.1 son pequeños y no llevarían el procesamiento en los extremos a un estado crítico que afecte el concepto de calidad del servicio (QoS). En el cliente como máximo el procesador se consumiría en un 5% durante menos de un segundo.

El uso de algoritmos de compresión para XML como XMLPPM a pesar de que comprime un 9% más que un algoritmo no específico para XML como gzip, consume el doble en procesamiento por el análisis que tiene que realizar. Se definió que a pesar de ocupar el doble de procesamiento que un algoritmo como gzip, no lleva a estado crítico por encima del 85% el uso del procesador, lo que no

afecta negativamente el tema de QoS, pero se estableció que el análisis de sintaxis en el caso de los clientes de clientes de Servicios Web en los PDAs queda resuelto por las propias librerías que en los dos ambientes, Pocket PC y Palm, generan los Stubs (los clientes) de los Servicios Web. En el caso del servidor, los servicios generados por Microsoft .NET Framework y Apache Axis, incluyen analizadores de sintaxis (parsers) XML. En Por otro lado, lo que se comprime al final es HTTP 1.1, y lo mejor es comprimir lo que al final va a viajar por la red. El comprimir antes el XML obligaría a una descompresión de lo generado por XMLPPM para poder emitir HTTP 1.1 por parte del servidor de Servicios Web. Con gzip, es en la última etapa, la etapa antes del envío de información, en la que se comprime el HTTP 1.1.

La creación de Servicios Web en los ambientes Microsoft Windows y Apache Axis, están suficientemente documentas en la red. Para los dispositivos Pocket PC existen numerosos documentos y asistentes que permiten generar clientes de estos servicios. Caso contrario ocurre con los dispositivos PDA para los cuales aunque existen excelentes IDEs para la creación de aplicaciones, no existen IDEs para la creación de clientes de Servicios Web, siendo la única posibilidad (donde no se posee un IDE particular, sino una serie de archivos y librerías) el proyecto gSOAP creado originalmente para clientes convencionales como PCs trabajando en C/C++. Este proyecto no se centra en PDAs, sólo genera algunas recomendaciones bastante sencillas para el caso d los PDAs. Tal es el caso del archivo stdsoap2.c que sobrepasa todos los límites de un PDA en cuanto a recursos de manejo de archivos y de enlace hacia funciones compiladas, tema que se resuelve de manera bastante manual, siguiendo las recomendaciones de secciones de código en la documentación de pre-tools en la programación para Palm.

Se ha observado que la fabricación de PDAs con procesamiento que llegan a 400Mhz es ya común. Se determinó que igualmente la tendencia es ofrecer el concepto total de movilidad que se logra cuando el dispositivo móvil puede conectarse a la red (GPRS en la mayoría de los casos). Esta conectividad es recurso crítico, y la compresión para ahorro de ancho de banda revierte bastante importancia en el concepto de movilidad.

REFERENCIAS

- TIA01: M. Tian; T. Voigt; T. Naumowicz; H. Ritter; J. Schiller, Performance Considerations for Mobile Web Services, 2003
- BAN01: V. Bansal; A. Dalton, A Performance Analysis of Web Services on Wireless PDAs,
- WON01: W. Wong, Exploring the boundaries of Web services on resource limited devices, 2003
- W3C01: World Wide Web Consortium, Web Services Activity, <http://www.w3.org/2002/ws/>
- PLM01: Palm, Inc., Integrating Mobile Data Services into Enterprise Infrastructures, 2003
- GSM01: Mobile Lifestreams Ltd, What is General Packet Radio Service, 2000, <http://www.gsmworld.com/technology/gprs/intro.shtml>
- OMA01: Open Mobile Alliance, Open Mobile Alliance Home Page, <http://www.openmobilealliance.org/>
- W3C02: World Wide Web Consortium, World Wide Web Consortium Home Page, <http://www.w3c.org>
- WSI01: Web Services Interoperability, Web Services Interoperability Home Page, <http://www.ws-i.org/>
- OAS01: Organization for the Advancement of Structured Information Standards, Organization for the Advancement of Structured Information Standards Home Page, <http://www.oasis-open.org/home/index.php>
- MIC01: Microsoft Corporation, .NET Framework Developer Center, <http://msdn.microsoft.com/netframework/>
- APC01: The Axis Development Team, Apache Axis Home Page - WebServices - Axis, <http://ws.apache.org/axis/>
- GAI01: J. Gailly; M. Addler, gzip Home Page, <http://www.gzip.org>
- PPM01: Sourceforge, XMLPPM: XML-Conscious PPM Compression, <http://www.cs.cornell.edu/People/jcheney/xmlppm/xmlppm.html>
- RDG01: Red-gate Software, ANTS Load Home Page, http://www.red-gate.com/dotnet/load_testing.htm
- BRL01: Borland, Optimizeit Enterprise Suite, <http://www.borland.com/optimizeit/>