

Del SIG de escritorio al entorno cliente-servidor con Web Processing Service

J. Masó¹, Xavier Pons^{2,1}

¹ Centre de Recerca Ecològica i Aplicacions Forestals (CREAF)
Universitat Autònoma de Barcelona (UAB)
Bellaterra (Barcelona)
{joan.maso}@uab.es

² Departament de Geografia
Universitat Autònoma de Barcelona (UAB)
Bellaterra (Barcelona)
{xavier.pons}@uab.es

Resumen

Esta ponencia discute algunas de las posibles arquitecturas que puede presentar un servicio WPS a partir de un conjunto de módulos disponibles en un SIG de escritorio. Se discute una arquitectura centrada en servidor como las usadas en la nueva web 2.0 y una arquitectura donde el servidor es un mero ejecutador de procesos basados en datos remotos. Cada arquitectura necesita de distintos elementos adicionales que no se describen en el propio estándar WPS, que son identificados y para los cuales se proponen soluciones.

Palabras clave: Proceso, estándar, servicio, OGC, WPS.

1 Introducción

La aplicación de protocolos estandarizados en los portales de cartografía en Internet ha permitido el acceso a usuarios de todo tipo y formación a la visualización y consulta de cartografía de referencia y temática. Sin embargo, el usuario experto, cuando trabaja en entornos *web*, echa en falta herramientas que le permitan procesar los datos disponibles, cruzarlos y obtener resultados que puedan

ser de ayuda a la toma de decisiones. La mayoría de estas herramientas requieren de la ejecución de procesos.

El estándar *Web Processing Service* (WPS) permite definir una interfaz estandarizada por el Open Geospatial Consortium (OGC) para la ejecución de procesos en un entorno cliente-servidor [4]. Estos procesos pueden ser, por ejemplo, algoritmos de análisis, cálculos geodésicos de cambio de sistema de referencia, modelos estadísticos aplicados a prospectiva territorial, etc, que son funcionalidades típicas de los verdaderos SIG de escritorio. Por ello parece indicado sugerir la extensión de las funcionalidades de un SIG de escritorio a un entorno cliente- servidor usando WPS. En su primera versión, aprobada en 2007, este estándar permite identificar los datos georeferenciados origen, iniciar el proceso, controlar el estado de ejecución del mismo y entregar o identificar los datos de salida.

El estándar deja totalmente abierto el origen de los datos pudiendo, pues, ser diverso: los datos se encuentran en el propio servidor, los datos son accesibles desde un servidor *Web Feature Service* (WFS) o *Web Coverage Service* (WCS), los datos pueden extraerse de una geodatabase, los datos están embebidos en la propia petición de ejecución como una colección de objetos *Geography Markup Language* (GML) o los datos están en el cliente y este debe enviarlos al servidor con anterioridad. Igualmente, este estándar WPS deja totalmente abierto el método a seguir para devolver los resultados de la ejecución al servidor pudiendo ser: almacenado en el propio servidor, mostrado al usuario a través de un servicio de generación de representaciones pictóricas y posterior entrega mediante *Web Map Service* (WMS) [1], ofrecido a través de un servicio de descarga (WFS, WCS, ftp), transferido a un servidor WFS transaccional, entregado como un conjunto de objetos GML [2] embebidos en la respuesta de la ejecución, o listados para su descarga posterior [3].

Por ello, un entorno que permita la ejecución de procesos WPS puede tener diversas arquitecturas en función de los objetivos del mismo. Esta ponencia plantea dos posibles arquitecturas a las que pueden optar los desarrolladores de programas SIG de escritorio que ya dispongan de una colección de algoritmos, procesos y módulos de calidad, para trabajar con información geográfica para desarrollar un servicio WPS. Las dos soluciones son analizadas desde el punto de vista del usuario, desde el punto de vista del rendimiento y desde el punto de vista de la dificultad de implementación. Por supuesto existen otras aproximaciones posibles, pero requieren de la participación de otros geoservicios, quedando más lejos del *software* de escritorio, que es el punto de partida de esta comunicación.

La primera arquitectura estudiada realiza una aproximación centrada en el servidor, siendo éste el almacén y el procesador, y limitando el uso del cliente a un simple visualizador de vistas de la información. Esta aproximación está muy de moda en el mundo de los nuevos servicios *Web 2.0*, donde servidores como *Flickr* permiten al usuario mantener en el centro de datos (*datacenter*) del servicio toda la información (en este caso fotografías) y, desde este, manipularla (procesos), clasificarla, documentarla (metadatos, en el fondo) y compartirla. Llamaremos a esta aproximación WPS-W2

La segunda aproximación mantiene la información en el cliente, por lo que el servidor es meramente una herramienta para la realización del proceso. Esta aproximación se parece más a la arquitectura del SIG de escritorio que puede tomar datos de diferentes ubicaciones a partir de la red local, procesarlos en la CPU del ordenador local y reenviarlos a un disco local o de red, sólo que en este caso la perspectiva se invierte y es el servidor WPS remoto el que tiene la capacidad de tomar datos de otras ubicaciones. Ni que decir tiene que esta aproximación requiere de un mayor ancho de banda en el tráfico de redes respecto a la primera aproximación. Llamaremos a esta aproximación WPS-R

2 Servidor de procesos en web WPS-W2

Las arquitecturas *Web 2.0* se basan en el poder de la interacción con el usuario como fuente de nuevos datos. Estos datos se almacenan en grandes centros de datos de ubicación desconocida, e inaccesibles directamente por el usuario. El usuario debe utilizar los recursos que el servicio le aporta o transferir sus datos al sistema con anterioridad a cualquier acción. Esto simplifica la arquitectura WPS dado que el usuario dispone de un catálogo de capas con las que trabajar que será idéntico sea cual sea la ubicación de dicho usuario. De igual manera, el resultado de una ejecución será almacenado en el propio centro de datos. El sistema dispondrá de mecanismos para la visualización del resultado de la ejecución del proceso y, eventualmente, para la exportación del resultado.

Para la ejecución del proceso, el usuario deberá seleccionar las capas implicadas de entre las disponibles en el catálogo del servicio y posiblemente completar algunos metadatos de la nueva capa generada. Una vez solicitado el inicio del proceso, el servicio lo ejecutará e incluirá el resultado en el catálogo una vez acabado el proceso. Sólo entonces el usuario podrá ver el resultado si lo desea.



Figura 1. Arquitectura WPS-W2

3 Servidor de procesos remoto WPS-R

Para que sea posible el uso de WPS-R es necesario que exista una interfaz de usuario en el lado cliente, un mecanismo para la transmisión de documentos cartográficos (en adelante *capas*, para simplificar) del cliente al servidor, una comunicación con el servidor para iniciar el proceso (junto con la información de los parámetros de entrada), un mecanismo para informar al usuario del estado del proceso y, finalmente, un sistema para transmitir al usuario el resultado del proceso, que generalmente será una capa pero puede ser una tabla o un solo valor. A continuación vamos a detallar cada uno de estos pasos.

La interfaz de usuario puede ser la misma usada por el programa de escritorio con la salvedad de que debe indicarse qué servidor debe realizar la operación. El

mecanismo de transmisión de capas del cliente al servidor puede no ser un tema de fácil solución para la mayoría de SIGs de escritorio dado que una capa puede estar formada por un gran número de archivos interrelacionados, o incluso una estructura de registros en varias tablas de una base de datos. Solamente algunos programas SIG disponen de mecanismos para la transmisión de una capa o un proyecto cartográfico en un solo archivo: ArcInfo dispone del formato de archivos *export* (e00), MiraMon dispone del formato de archivo MMZ y el reciente Google Earth dispone del formato de archivo KMZ. Este tipo de archivos (todos ellos con capacidad de compresión) son ideales para transmitir capas del cliente al servidor manteniendo la mayoría o toda (según el formato) la calidad de los datos originales (geometría, geodesia, datos espaciales y alfanuméricos, tablas y documentos relacionados, metadatos, simbolización, etc). En adelante, llamaremos a estos formatos, *formatos de intercambio*. Cada servidor WPS se basa en un perfil concreto del estándar WPS que establece claramente cómo iniciar la ejecución de un proceso y como relacionar las capas para cada proceso disponible. Una vez iniciado el proceso, la aplicación cliente puede consultar el estado de la ejecución al servidor y actualizar una ventana de información al usuario. Si el final de la ejecución da como resultado una capa, antes de dar por finalizado el proceso la aplicación servidora deberá empaquetar el resultado en un formato de intercambio para su entrega. Una vez la aplicación cliente conozca que el proceso ha terminado deberá solicitar la descarga del archivo de intercambio resultante y, si lo estima oportuno, mostrarlo al usuario final.

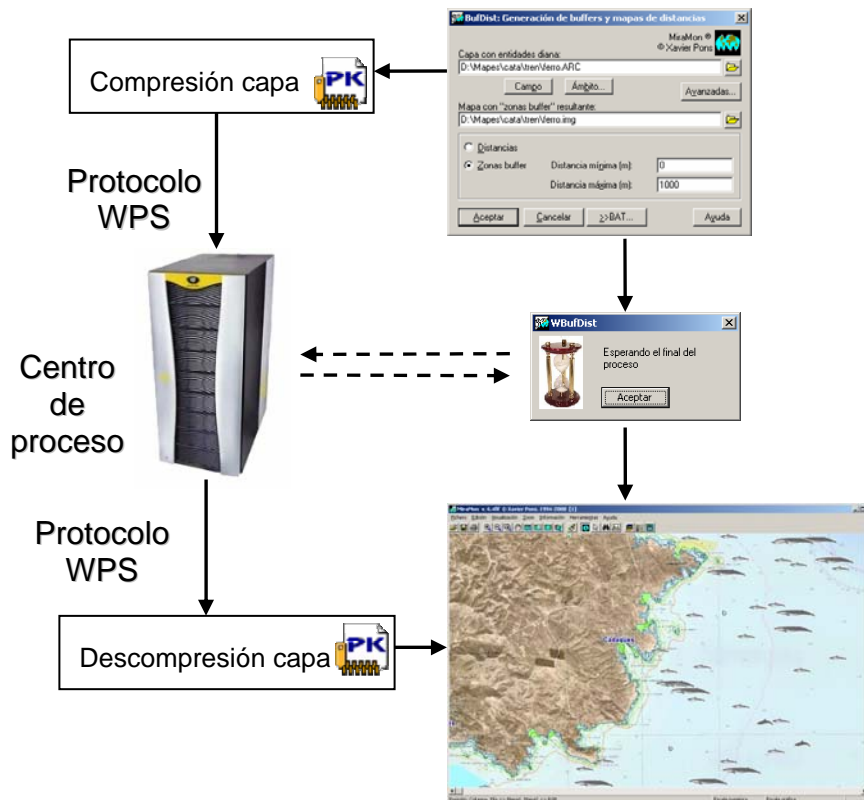


Figura 2. Arquitectura WPS-R

Un punto práctico importante es que en el caso de que el servidor y el cliente se encuentren en el mismo dominio y que el esquema de permisos así lo permita (por ejemplo cuando ambos se encuentren en la misma red de área local), los pasos de compresión de las capas a procesar así como la descompresión de la capa de salida pueden ser eliminados dado que se supone que el servidor tiene acceso a los mismos datos que el usuario posee y que el cliente puede acceder directamente al producto resultante de la operación ejecutada.

4 Ejemplo

A modo de ejemplo, mostramos aquí el prototipo del módulo WBufDist de MiraMon siguiendo la arquitectura WPS-R. Todos los módulos de MiraMon están formados por un cuadro de diálogo independiente de la ejecución en sí misma, la cual es realizada por una aplicación de tipo consola. Ambos elementos se

comunican a través de una sintaxis en línea de comando basada en los procesos BATCH de la consola de Windows. El cuadro de diálogo de MiraMon de escritorio ha sido adaptado para WPS. El usuario sólo percibe que un nuevo parámetro debe ser introducido para indicar que la ejecución será realizada por un servidor externo como se muestra en la figura 3. Por otro lado, el botón “BAT>>”, siempre presente en las cajas de funcionalidades de MiraMon, ha cambiado su comportamiento y ahora muestra la sintaxis XML de la petición *Execute* de WPS.

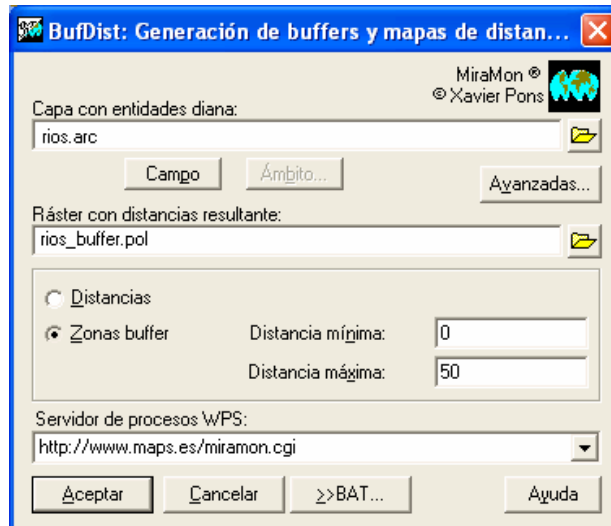


Figura 3. \$\$ ATENCIÓN: TEXT A REVISAR: Fragmento del documento XML de respuesta de la operación de buffer vectorial con el vínculo a la capa comprimida resultado

Por otro lado, la aplicación de tipo consola ha sido convertida en una aplicación tipo cgi que reside en un servidor web y que puede ser llamada remotamente por el cuadro de diálogo *o por cualquier aplicación externa*. Así, por ejemplo, la descripción de la operación de buffer vectorial puede ser obtenida invocando el comando DescribeProcess como se muestra en la figura 4 y 5.

```
http://www.maps.es/miramon.cgi?Service=WPS&
Request=DescribeProcess& Version=1.0.0& Language=ca-CA$$
Identifier=InputVectorTopo
```

Figura 4. Petición de la descripción de la operación de buffer vectorial

```

<wps:ProcessDescriptions>
  <ProcessDescription wps:processVersion="2"
    storeSupported="true" statusSupported="false">
    <ows:Identifier>BufferVectorial</ows:Identifier>
    <ows:Title>Creación de un buffer vectorial sobre
    puntos, líneas o polígonos.</ows:Title>
    <ows:Abstract>Crea un buffer alrededor de una ...
    <ows:Metadata xlink:title="buffer" />
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>InputVectorTopo</...>
        <ComplexData>
          <Default><Format>
            <MimeType>application/x-mmz</MimeType>
            <Encoding>binary</Encoding>
          </Format></Default>
        </ComplexData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>ZonaBufferMax</ows:Identifier>
        <LiteralData>
          <ows:DataType>float</ows:DataType>
          <UOMs><Default>
            <ows:UOM>meters</ows:UOM>
          </Default></UOMs>
          <DefaultValue>100</DefaultValue>
        </LiteralData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>ZonaBufferMin</ows:Identifier>...
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>BufferedVector</ows:Identifier>
        <ComplexOutput>

```



```

    <Default><Format>
      <MimeType>application/x-mmz</MimeType>
      <Encoding>binary</Encoding>
    </Format></Default>
  </ComplexOutput>
</Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>

```

Figura 5. Fragmento de la descripción de la operación de buffer vectorial. Es la respuesta de la petición de la figura 4

El resultado es devuelto al cliente como un archivo de polígonos comprimido en un archivo MMZ que se encuentra vinculado por un documento XML de respuesta. En el caso de que se realice una petición a un proceso de ejecución que requiera de un tiempo de espera largo, el estándar prevé una respuesta inmediata de un XML con un vínculo al futuro resultado dejando en manos de la aplicación cliente la responsabilidad de consultar cada cierto tiempo la ubicación indicada hasta encontrar el archivo esperado creado y listo para su descarga.

```

<wps:ExecuteResponse>
  <wps:Process wps:processVersion="1">
    <ows:Identifier>BufferVectorial</ows:Identifier>
  </wps:Process>
  <wps:Status creationTime="2008-11-02T12:13:14Z">
    <wps:ProcessSucceeded/>
  </wps:Status>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>InputVectorTopo</ows:Identifier>
      <wps:Reference xlink:href="http://www.maps.es/rios.mmz"
        mimeType="application/x-mmz" encoding="binary"/>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>ZonaBufferMax</ows:Identifier>
      <wps>Data>

```

```

        <wps:LiteralData uom="meters">50</wps:LiteralData>
    </wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:OutputDefinitions>
    <wps:Output mimeType="application/x-mmz"
asReference="true">
        <ows:Identifier>BufferedVector</ows:Identifier>
    </wps:Output>
</wps:OutputDefinitions>
<wps:ProcessOutputs>
    <wps:Output>
        <ows:Identifier>BufferedVector</ows:Identifier>
        <wps:Reference href="http://www.maps.es/rios_buffer.mmz"
mimeType="application/x-mmz"/>
    </wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>

```

Figura 6. Fragmento del documento XML de respuesta de la operación de buffer vectorial con el vínculo a la capa comprimida resultado

5 Conclusiones

Los protocolos de comunicaciones y navegadores actuales, así como la velocidad de las redes, permite el despliegue de servidores WPS con diferentes arquitecturas. La aparición de dispositivos con pocas capacidades de proceso pero con capacidad de comunicación más que razonable, como las PDA y los teléfonos móviles inteligentes, justifican la delegación de determinados procesos a servidores externos. Por otro lado, la arquitectura que hemos denominado WPS-R tiene la ventaja que permite la evolución del SIG de escritorio a un SIG basado en WPS con unas mínimas modificaciones en el entorno de diálogo del usuario, aunque aumenta el tráfico de red, por lo que sólo es posible aplicarlo en entornos de banda ancha, si las capas son mínimamente voluminosas. En los entornos de banda estrecha, tiene justificación el uso de los servicios *web* de procesos donde los datos residan en el servidor y un entorno web permita tanto su edición como su procesado a partir de una arquitectura que hemos llamado WPS-W2.

Referencias

- [1] de la Beaujardiere, Jeff (ed.): OpenGIS® Web Map Server Implementation Specification 1.3.0, OGC Document OGC 06-042, <http://www.opengeospatial.org/standards/wms>, (2006).
- [2] Cox, Simon et. al. (ed.): OpenGIS® Geography Markup Language (GML) Implementation Specification 3.1.0, OGC Document OGC 03-105r1, <http://www.opengeospatial.org/standards/gml>, (2003).
- [3] Peng, Zhong-Ren y Ming-Hsiang Tsou: Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks. Wiley, (2003).
- [4] Peter Schut, Petter. (ed.): OpenGIS® Web Processing Service 1.0.0, OGC Document OGC 05-007r7, <http://www.opengeospatial.org/standards/wps>, (2007).