

Tipo de artículo: Artículo original
Temática: Soluciones Informáticas
Recibido: 01/03/18 | Aceptado: 06/04/18 | Publicado: 27/04/18

Componente para brindar servicios basados en la localización utilizando computación distribuida

Component to provide location based services using distributed programming

Ofir Alfonso Cantillo¹

¹Facultad de Ingeniería Informática, Universidad Tecnológica de la Habana “José Antonio Echeverría”, oalfonso@ceis.cujae.edu.cu

* Autor para correspondencia: oalfonso@ceis.cujae.edu.cu

Resumen

Con el transcurso del tiempo, debido a la gran cantidad de datos digitales con que se cuenta, se ha convertido en una necesidad el procesamiento de la información utilizando variados sistemas informáticos. Grandes cantidades de datos cuentan con un componente espacial por lo que se necesitan sistemas que comprendan ese tipo de información y sean capaces de procesarla. Con ese objetivo es que surgen los Sistemas de Información Geográfica los cuales con el tiempo han evolucionados hacia las Infraestructuras de Datos Espaciales. Las Infraestructuras de Datos Espaciales se encargan de brindar diversos servicios entre los que se encuentran los Servicios Basados en la Localización. Existen varias herramientas que pueden ser utilizadas con el fin de brindar estos servicios, entre las que se encuentra el servidor de mapas GeoServer. GeoServer cuenta con un componente capaz de brindar Servicios Basados en la Localización, pero los tiempos de respuesta de estos servicios para petición de grandes cantidades de direcciones es significativamente alto. El objetivo del presente trabajo es disminuir los tiempos de respuestas del componente gs-opens. Para ello se van a utilizar técnicas de computación paralela y distribuida que sean compatibles con los estándares del servidor GeoServer. Con el diseño de experimentos llevado a cabo, se compararon los tiempos de respuestas de los dos sistemas: secuencial y paralelo. Se demostró mediante pruebas estadísticas que el tiempo de respuesta de la versión distribuida del componente gs-opens es significativamente menor que los tiempos de respuestas del componente diseñado con programación secuencial.

Palabras clave: GeoServer; Computación paralela; Servicios Basados en la Localización; Sistemas de Información Geográfica.

Abstract

Over the years, due to the large amount of digital data available, it has become a necessity to process information using many computer systems. Large amounts of data have a spatial component, so it's a necessity the development of systems that understand this kind of information and to be capable of processing it. With this goal, Geographic Information Systems emerge, which over time have evolved into Spatial Data Infrastructures. The Spatial Data Infrastructures are responsible for providing some kind of services among which are Location Based Services that use the location to provide relative information to users. There are several tools that can be used in order to provide these services, among which is GeoServer map server. GeoServer has a component capable of providing Location Based Services, but the response times of these services for requesting large numbers of addresses is significantly high. The objective of this paper is to reduce the response times of the gs-opens component. For this, parallel and distributed computing techniques that are compatible with GeoServer server standards will be used. With the design of experiments carried out once the component was developed, the response times of the two systems were compared: sequential and parallel. It was demonstrated by statistical tests that the response time of the distributed version of the gs-opens component is significantly less than the response times of the component designed with sequential programming.

Keywords: *Geographic Information Systems; GeoServer; Location Based Services; Parallel Computing.*

Introducción

El uso de la información geográfica en el día a día de las personas ha tomado un auge significativo en los últimos años. Servicios populares de internet, como Google Maps, Bing Maps y OpenStreetMap, que tienen como base el uso de datos geográficos, han provisto a la población de herramientas atractivas para la consulta y manipulación de información espacial.

El continuo acceso de las personas a la información y el desarrollo de las tecnologías han provocado un aumento de la información digital en la que parte de sus componentes poseen información espacial [1]. Para hacer uso de la información espacial es necesario contar con un Sistema de Información Geográfica (SIG) definido como:

“un SIG es un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión” [2].

Debido al amplio crecimiento experimentado por este tipo de información y a la cantidad de usuarios que han comenzado a hacer uso de esta, los SIG han evolucionado a instancias superiores, conocidas como Infraestructuras de Datos Espaciales (IDE). Una IDE es una colección de tecnologías, políticas y disposiciones institucionales que facilitan el acceso a datos espaciales [3].

Con el crecimiento de los volúmenes de información que es almacenada día a día, se hace necesario contar con recursos capaces de su procesamiento y aprovechar al máximo aquellos con los que se cuenta. Para resolver problemas donde se necesita un gran poder de cómputo surge la computación distribuida. La computación distribuida es el uso simultáneo de múltiples recursos de cómputo para resolver un problema computacional donde se necesite, entre otras cosas, analizar o procesar grandes cantidades de datos [4].

Las IDE se componen, además de lo planteado anteriormente, por diversos servicios estandarizados entre los que se encuentran el Servicio Web de Objetos Geográficos (WFS, por sus siglas en inglés) [5] el Servicio Web de Mapas (WMS, por sus siglas en inglés) [6] y el Servicio Web de Catálogo (WCS, por sus siglas en inglés) [7], por solo citar algunos, pero los servicios basados en la localización (LBS, por sus siglas en inglés) no ha alcanzado un nivel similar a los otros servicios mencionados [1].

Los Servicios Basados en la Localización (LBS), en su forma estandarizada (OpenLS) especifican estándares que permiten a las compañías y usuarios en el sector de LBS proveer a sus aplicaciones (respuesta de emergencia, navegador personal, servicio de proximidad, servicio de localización, entre otras) con interfaces que provean los servicios necesarios para sus propósitos: Servicio de Geocodificación, Servicio de Directorio, Servicio de Ruta, entre otros [8].

En [9] se definió el componente `gs-opensls` para GeoServer capaz de proporcionar los LBS en el entorno cubano. GeoServer es un servidor de código abierto escrito en Java que permite a los usuarios compartir y editar información espacial. Destinado para la interoperabilidad, es capaz de publicar datos de gran variedad de fuentes usando estándares [10].

Según [9] el componente `gs-opensls` proporciona soporte a la geocodificación en lotes que no es más que la capacidad de procesar muchas direcciones en peticiones de varias cantidades y es precisamente la cantidad de direcciones un factor que influye negativamente en el tiempo de respuesta del servicio de geocodificación. El flujo de trabajo del componente `gs-opensls` en su versión secuencial se muestra en la Fig. 1.

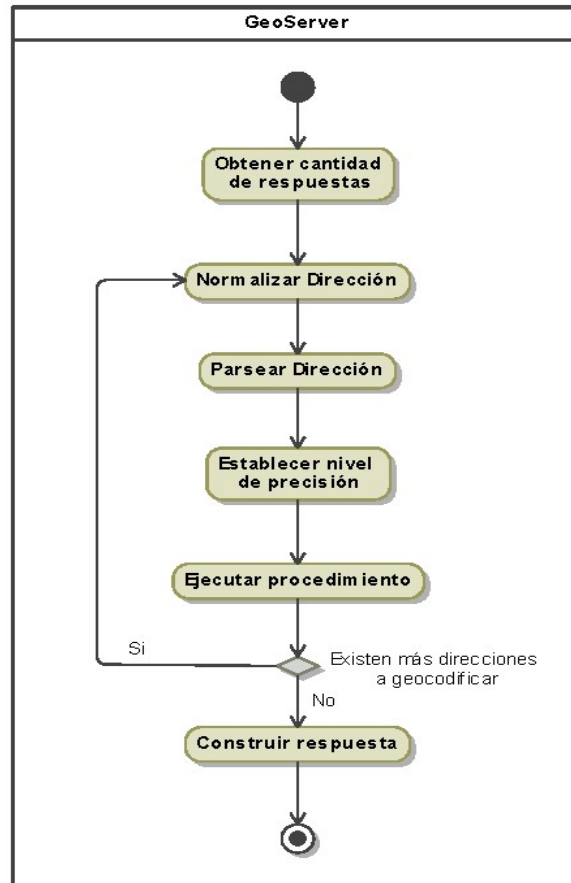


Fig. 1: Versión secuencial del componente gs-openls

A partir de la situación antes descrita se define como problema ¿cómo disminuir el tiempo de respuesta del servicio de geocodificación de direcciones postales para GeoServer?, siendo así el objetivo principal de este trabajo la disminución de los tiempos de respuesta del servicio de geocodificación de direcciones postales.

La solución propuesta consiste en implementar el componente gs-openls utilizando técnicas de computación distribuida.

Materiales y métodos o Metodología computacional

GeoServer es un servidor de mapas que utiliza estándares para el intercambio de la información espacial entre el cliente y el servidor.

El componente gs-opensls fue desarrollado para ser integrado a GeoServer. Este servidor está implementado sobre el lenguaje Java con el objetivo de aprovechar la flexibilidad e independencia del sistema operativo que ofrece. En [9] se define la versión 8 de Java como el lenguaje a utilizar para desarrollar el componente gs-opensls.

Fue utilizado el gestor de proyectos Apache Maven con el objetivo de gestionar todas las dependencias que solicita GeoServer, además, fue utilizado para compilar el componente e incluirlo a GeoServer.

IntelliJ IDEA es un software escrito en java que permite la integración de diversas tecnologías con el objetivo de desarrollar un proyecto informático. Permite la integración de GeoServer con Apache Maven por lo que facilita la gestión de extensiones para el servidor de mapas.

OpenLS Core es una biblioteca que contiene todas las clases que describen los elementos que se encuentran en la especificación OpenLS de OGC. Es utilizada para la interpretación de las peticiones de los usuarios al servidor y para construir la respuesta de geocodificación que le será enviada al cliente.

Minitab16 es el software utilizado para la ejecución de las pruebas estadísticas realizadas al software.

Para la comunicación del componente con las bases de datos necesarias para el proceso de geocodificación de direcciones postales se utilizaron las bibliotecas OJDBC y PGJDBC para la conexión con los procedimientos almacenados en las bases de datos Oracle y PostgreSQL respectivamente.

GeoServer es un servidor de mapas que utiliza estándares para el intercambio de la información espacial entre el cliente y el servidor.

El componente gs-opensls fue desarrollado para ser integrado a GeoServer. Este servidor está implementado sobre el lenguaje Java con el objetivo de aprovechar la flexibilidad e independencia del sistema operativo que ofrece. En [9] se define la versión 8 de Java como el lenguaje a utilizar para desarrollar el componente gs-opensls.

Fue utilizado el gestor de proyectos Apache Maven con el objetivo de gestionar todas las dependencias que solicita GeoServer, además, fue utilizado para compilar el componente e incluirlo a GeoServer.

IntelliJ IDEA es un software escrito en java que permite la integración de diversas tecnologías con el objetivo de desarrollar un proyecto informático. Permite la integración de GeoServer con Apache Maven por lo que facilita la gestión de extensiones para el servidor de mapas.

OpenLS Core es una biblioteca que contiene todas las clases que describen los elementos que se encuentran en la especificación OpenLS de OGC. Es utilizada para la interpretación de las peticiones de los usuarios al servidor y para construir la respuesta de geocodificación que le será enviada al cliente.

Minitab16 es el software utilizado para la ejecución de las pruebas estadísticas realizadas al software.

Para la comunicación del componente con las bases de datos necesarias para el proceso de geocodificación de direcciones postales se utilizaron las bibliotecas OJDBC y PGJDBC para la conexión con los procedimientos almacenados en las bases de datos Oracle y PostgreSQL respectivamente.

Resultados y discusión

Se realizó un estudio del componente de geocodificación desarrollado con anterioridad con el objetivo de evaluar sus características, estructura y funcionamiento para utilizarlo como punto de partida en el desarrollo de un nuevo componente de geocodificación de direcciones postales cubanas utilizando computación distribuida. El objetivo de aplicar técnicas de computación distribuida en el desarrollo de un componente para la geocodificación es el de disminuir los tiempos de respuesta del servidor cuando recibe una petición que contenga varios objetos a geocodificar.

Una petición con estas características se muestra en la Fig. 2:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<XLS version="1.2" xmlns="http://www.opengis.net/xls">
  <RequestHeader/>
  <Request maximumResponses="10"
    methodName="GeocodeService"
    requestID="" version="1.2">
    <GeocodeRequest>
      <Address>
        <freeFormAddress> 60 # 109 APTO 2 E/ AVE 1 A Y
          AVE 1, MIRAMAR, PLAYA, CIUDAD DE LA HABANA
        </freeFormAddress>
      </Address>
      <Address>
        <freeFormAddress>340 E/ 107 Y 109, RPTO ALT
          NARANJAL, MUNIC, MATANZAS, MATANZAS
        </freeFormAddress>
      </Address>
    </GeocodeRequest>
  </Request>
</XLS>
```

Fig. 2: Petición de geocodificación de múltiples direcciones

Cuando GeoServer recibe una petición como la mostrada en el código anterior geocodifica en un primer instante la dirección “60 # 109 Apto 2 E/ Ave 1 A Y Ave 1, Miramar, Playa, La Habana” y cuando finaliza ese proceso comienza la geocodificación de la dirección “340 E/ 107 Y 109, RptoAlt Naranjal, Munic Matanzas, Matanzas”. Esto provoca retraso en la respuesta al cliente.

Implementado la técnica de computación distribuida conocida como “Memoria Compartida” se puede paralelizar el

procesamiento de los datos (paralelismo de datos) y así disminuir los tiempos de respuesta del servidor. En [4] se define como sistemas que utilizan memoria compartida a aquellos donde sus elementos de procesamiento comparten un mismo espacio de memoria para realizar sus funciones.

En [11] se definen varias técnicas incluidas en el lenguaje Java 8 para hacer uso de la computación distribuida y que pueden ayudar a resolver el problema planteado. Implementado la clase Callable se puede hacer uso de la técnica de computación distribuida conocida como Memoria Compartida. Cuando cada elemento de procesamiento culmina sus objetivos se puede obtener el resultado del procesamiento de cada hilo utilizando la clase Future para, posteriormente, confeccionar el XML de respuesta y enviárselo al cliente. Un ejemplo de respuesta se ofrece en la Fig. 3:

```
<GeocodeResponseList numberOfGeocodedAddresses="1">
  <GeocodedAddress>
    <Address>
      <freeFormAddress>Calle 60 entre Avenida 1
        y Avenida 1 A, Miramar, Playa, Ciudad de La Habana
      </freeFormAddress>
    </Address>
    <Point srsName="EPSG:4326">
      <pos>-82.435526 23.116071</pos>
    </Point>
  </GeocodedAddress>
</GeocodeResponseList>
<GeocodeResponseList numberOfGeocodedAddresses="5">
  <GeocodedAddress>
    <Address>
      <freeFormAddress>Calle 346 entre Calle 107 A y
        Calle 109, Alturas del Naranjal, Matanzas
      </freeFormAddress>
    </Address>
    <GeocodeMatchCode/>
    <Point srsName="EPSG:4326">
      <pos>-81.598757 23.037505</pos>
    </Point>
  </GeocodedAddress>
</GeocodeResponseList>
```

Fig. 3: Respuesta de geocodificación de múltiples direcciones

Como se puede observar en el código anterior la respuesta al cliente está compuesta de dos etiquetas GeocodeResponseList dentro de las cuales se encuentran las direcciones geocodificadas para cada una de las 2 direcciones enviadas al servidor para geocodificarlas. Estas direcciones no dependen una de la otra para su geocodificación por lo que este proceso se pudiera desarrollar en paralelo.

El componente desarrollado geocodifica varias direcciones enviadas en una sola petición. De acuerdo a la arquitectura

del sistema donde se esté ejecutando GeoServer se crea una determinada cantidad de procesos en correspondencia con la cantidad de elementos de procesamientos con que se cuenta. De acuerdo a la cantidad de procesos creados, se distribuyen las direcciones a geocodificar de una forma balanceada por cada proceso. Una vez culminado el proceso de geocodificación para cada dirección, se confecciona un XML de respuesta que le será enviado al cliente. Tanto el XML de petición como el de respuesta tienen que seguir el estándar OpenLS. En la Fig.4. se muestra un diagrama de actividades donde se evidencia el flujo de trabajo del componente gs-opensls utilizando computación distribuida.

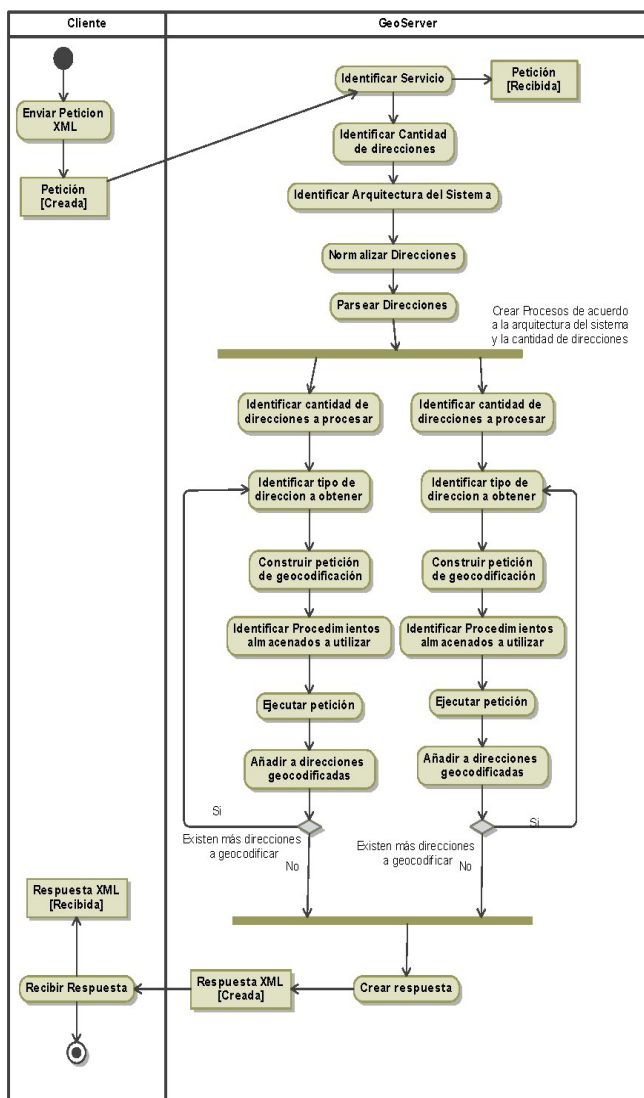


Fig. 4: Flujo de trabajo del componente gs-opensls utilizando computación paralela.

Una vez concluida el desarrollo del componente se realiza un proceso de evaluación de la solución desarrollada. El proceso de pruebas a la aplicación se realiza con el objetivo de validar la solución propuesta. Después del desarrollo de la aplicación se hace necesario el diseño de experimentos para verificar si la implementación distribuida del sistema reduce los tiempos de geocodificación de la implementación secuencial.

El proceso a analizar es el envío y recepción de peticiones al servicio de geocodificación donde se define como rendimiento del proceso el tiempo de respuesta del servicio.

A continuación, se definen los factores controlables identificados:

- Cantidad de direcciones enviadas: Es la cantidad de direcciones que se envía en el fichero de la petición al servicio de geocodificación.
- Cantidad de direcciones recibidas: Es la cantidad de direcciones que se reciben en el fichero de la respuesta que se obtiene del servicio de geocodificación para cada dirección solicitada.
- Cantidad de ejecuciones (Ciclos): Es la cantidad de veces que se repetirá la prueba. Al recibir la respuesta a una petición, se envía al sistema la nueva petición correspondiente a la siguiente ejecución.

Dentro de los factores no controlables se señalan:

- Tráfico de la red: Las fuentes de datos se encuentran ubicadas en unidades en la red del centro donde se ejecutan los experimentos. En el momento en que se realizan las peticiones no se puede controlar el tráfico que existe en la red.
- Procesos del sistema operativo: En el momento en que se realizan las peticiones al servidor hay procesos ejecutándose en el sistema operativo del servidor que consumen recursos arbitrariamente y no pueden ser detenidos.

La unidad experimental sobre la cual se ejecutan las pruebas es donde se encuentra alojado el servidor de mapas GeoServer que es el contenedor de los servicios. Las características se muestran en la tabla siguiente:

De acuerdo a los factores controlables del experimento se define como la cantidad de direcciones enviadas en una petición como 10000, la cantidad de direcciones máxima de respuesta serán 10 para cada una de las direcciones enviadas y se realizarán 10 réplicas de la prueba.

Los resultados de la ejecución de la prueba utilizando la versión secuencial y la versión distribuida del sistema se muestran en la Tabla 1.

Teniendo en cuenta los resultados obtenidos con la prueba realizada se puede observar que los tiempos de geocodificación de las 10000 direcciones es considerablemente menor en el sistema que utiliza técnicas de computación distribuida y paralela en comparación con la versión secuencial del sistema.

TABLA 1

Petición	Secuencial (Tiempo en minutos)	Distribuido (Tiempo en minutos)
1	135,153695	16,485268
2	112,368795	22,631273
3	196,365815	15,351222
4	184,954235	22,403083
5	106,847239	20,365843
6	115,487962	18,040153
7	181,451253	25,508933
8	194,339876	24,632589
9	160,630109	24,985624
10	189,583492	22,257634

Se realizó un estudio del componente de geocodificación desarrollado con anterioridad con el objetivo de evaluar sus características, estructura y funcionamiento para utilizarlo como punto de partida en el desarrollo de un nuevo componente de geocodificación de direcciones postales cubanas utilizando computación distribuida. El objetivo de aplicar técnicas de computación distribuida en el desarrollo de un componente para la geocodificación es el de disminuir los tiempos de respuesta del servidor cuando recibe una petición que contenga varios objetos a geocodificar.

Una petición con estas características se muestra en la Fig. 2:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<XLS version="1.2" xmlns="http://www.opengis.net/xls">
  <RequestHeader/>
  <Request maximumResponses="10"
    methodName="GeocodeService"
    requestID="" version="1.2">
    <GeocodeRequest>
      <Address>
        <freeFormAddress> 60 # 109 APTO 2 E/ AVE 1 A Y
          AVE 1, MIRAMAR, PLAYA, CIUDAD DE LA HABANA
        </freeFormAddress>
      </Address>
      <Address>
        <freeFormAddress>340 E/ 107 Y 109, RPTO ALT
          NARANJAL, MUNIC, MATANZAS, MATANZAS
        </freeFormAddress>
      </Address>
    </GeocodeRequest>
  </Request>
</XLS>
```

Fig. 2: Petición de geocodificación de múltiples direcciones

Cuando GeoServer recibe una petición como la mostrada en el código anterior geocodifica en un primer instante la dirección “60 # 109 Apto 2 E/ Ave 1 A Y Ave 1, Miramar, Playa, La Habana” y cuando finaliza ese proceso comienza la geocodificación de la dirección “340 E/ 107 Y 109, RptoAlt Naranjal, Munic Matanzas, Matanzas”. Esto provoca retraso en la respuesta al cliente.

Implementado la técnica de computación distribuida conocida como “Memoria Compartida” se puede paralelizar el procesamiento de los datos (paralelismo de datos) y así disminuir los tiempos de respuesta del servidor. En [4] se define como sistemas que utilizan memoria compartida a aquellos donde sus elementos de procesamiento comparten un mismo espacio de memoria para realizar sus funciones.

En [11] se definen varias técnicas incluidas en el lenguaje Java 8 para hacer uso de la computación distribuida y que pueden ayudar a resolver el problema planteado. Implementado la clase Callable se puede hacer uso de la técnica de computación distribuida conocida como Memoria Compartida. Cuando cada elemento de procesamiento culmina sus objetivos se puede obtener el resultado del procesamiento de cada hilo utilizando la clase Future para, posteriormente, confeccionar el XML de respuesta y enviárselo al cliente. Un ejemplo de respuesta se ofrece en la Fig. 3:

```
<GeocodeResponseList numberOfGeocodedAddresses="1">
  <GeocodedAddress>
    <Address>
      <freeFormAddress>Calle 60 entre Avenida 1
        y Avenida 1 A, Miramar, Playa, Ciudad de La Habana
      </freeFormAddress>
    </Address>
    <Point srsName="EPSG:4326">
      <pos>-82.435526 23.116071</pos>
    </Point>
  </GeocodedAddress>
</GeocodeResponseList>
<GeocodeResponseList numberOfGeocodedAddresses="5">
  <GeocodedAddress>
    <Address>
      <freeFormAddress>Calle 346 entre Calle 107 A y
        Calle 109, Alturas del Naranjal, Matanzas
      </freeFormAddress>
    </Address>
    <GeocodeMatchCode/>
    <Point srsName="EPSG:4326">
      <pos>-81.598757 23.037505</pos>
    </Point>
  </GeocodedAddress>
</GeocodeResponseList>
```

Fig. 3: Respuesta de geocodificación de múltiples direcciones

Como se puede observar en el código anterior la respuesta al cliente está compuesta de dos etiquetas GeocodeResponseList dentro de las cuales se encuentran las direcciones geocodificadas para cada una de las 2 direcciones enviadas al servidor para geocodificarlas. Estas direcciones no dependen una de la otra para su geocodificación por lo que este proceso se pudiera desarrollar en paralelo.

El componente desarrollado geocodifica varias direcciones enviadas en una sola petición. De acuerdo a la arquitectura del sistema donde se esté ejecutando GeoServer se crea una determinada cantidad de procesos en correspondencia con la cantidad de elementos de procesamientos con que se cuente. De acuerdo a la cantidad de procesos creados, se distribuyen las direcciones a geocodificar de una forma balanceada por cada proceso. Una vez culminado el proceso de geocodificación para cada dirección, se confecciona un XML de respuesta que le será enviado al cliente. Tanto el XML de petición como el de respuesta tienen que seguir el estándar OpenLS. En la Fig.4. se muestra un diagrama de actividades donde se evidencia el flujo de trabajo del componente gs-opensls utilizando computación distribuida.

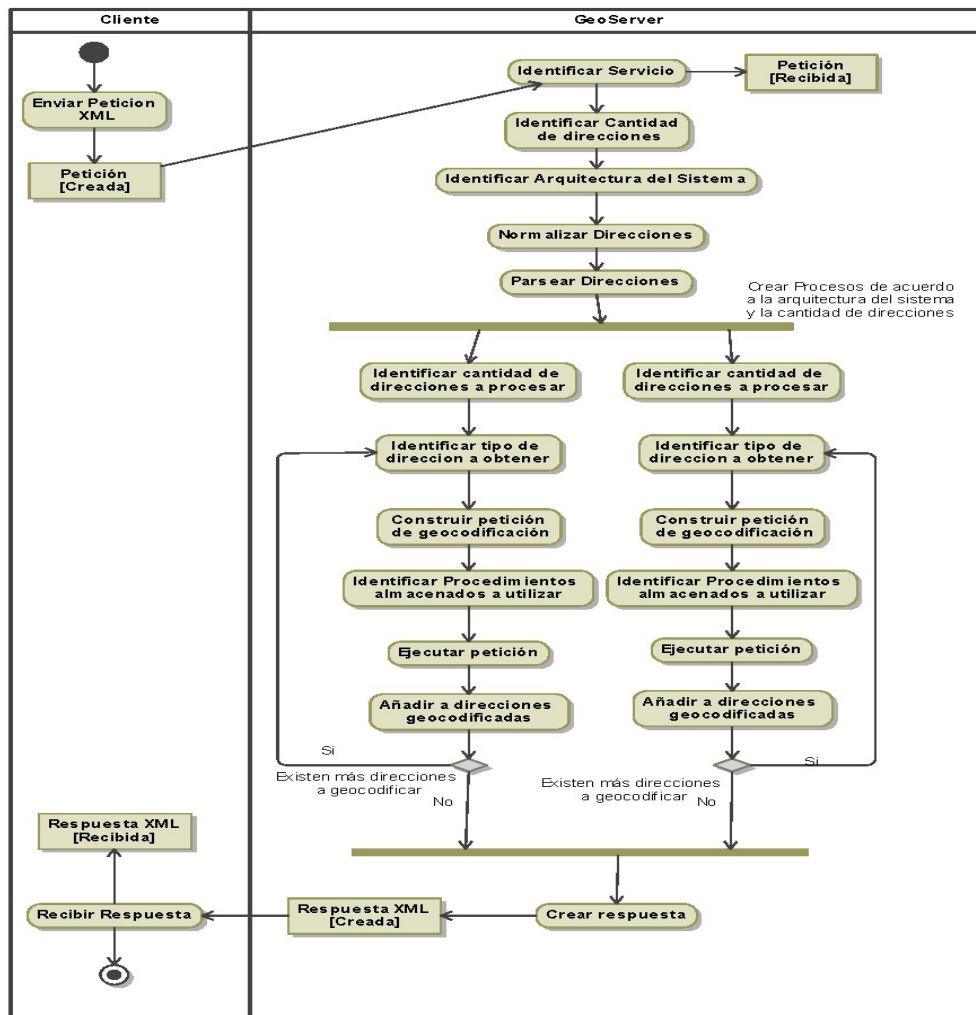


Fig. 4: Flujo de trabajo del componente gs-openls utilizando computación paralela.

Una vez concluida el desarrollo del componente se realiza un proceso de evaluación de la solución desarrollada. El proceso de pruebas a la aplicación se realiza con el objetivo de validar la solución propuesta. Después del desarrollo de la aplicación se hace necesario el diseño de experimentos para verificar si la implementación distribuida del sistema reduce los tiempos de geocodificación de la implementación secuencial.

El proceso a analizar es el envío y recepción de peticiones al servicio de geocodificación donde se define como rendimiento del proceso el tiempo de respuesta del servicio.

A continuación, se definen los factores controlables identificados:

- Cantidad de direcciones enviadas: Es la cantidad de direcciones que se envía en el fichero de la petición al servicio de geocodificación.
- Cantidad de direcciones recibidas: Es la cantidad de direcciones que se reciben en el fichero de la respuesta que se obtiene del servicio de geocodificación para cada dirección solicitada.
- Cantidad de ejecuciones (Ciclos): Es la cantidad de veces que se repetirá la prueba. Al recibir la respuesta a una petición, se envía al sistema la nueva petición correspondiente a la siguiente ejecución.

Dentro de los factores no controlables se señalan:

- Tráfico de la red: Las fuentes de datos se encuentran ubicadas en unidades en la red del centro donde se ejecutan los experimentos. En el momento en que se realizan las peticiones no se puede controlar el tráfico que existe en la red.
- Procesos del sistema operativo: En el momento en que se realizan las peticiones al servidor hay procesos ejecutándose en el sistema operativo del servidor que consumen recursos arbitrariamente y no pueden ser detenidos.

La unidad experimental sobre la cual se ejecutan las pruebas es donde se encuentra alojado el servidor de mapas GeoServer que es el contenedor de los servicios. Las características se muestran en la tabla siguiente:

De acuerdo a los factores controlables del experimento se define como la cantidad de direcciones enviadas en una petición como 10000, la cantidad de direcciones máxima de respuesta serán 10 para cada una de las direcciones enviadas y se realizarán 10 réplicas de la prueba.

Los resultados de la ejecución de la prueba utilizando la versión secuencial y la versión distribuida del sistema se muestran en la Tabla 1.

Teniendo en cuenta los resultados obtenidos con la prueba realizada se puede observar que los tiempos de geocodificación de las 10000 direcciones es considerablemente menor en el sistema que utiliza técnicas de computación distribuida y paralela en comparación con la versión secuencial del sistema.

Tabla 1. Resultado de la prueba

Petición	Secuencial (Tiempo en minutos)	Distribuido (Tiempo en minutos)
1	135,153695	16,485268
2	112,368795	22,631273
3	196,365815	15,351222

4	184,954235	22,403083
5	106,847239	20,365843
6	115,487962	18,040153
7	181,451253	25,508933
8	194,339876	24,632589
9	160,630109	24,985624
10	189,583492	22,257634

Con el objetivo de demostrar que la variante paralela mejora el tiempo de ejecución de la variante secuencial y establecer comparaciones entre ellas se utilizarán pruebas estadísticas.

Dentro de las pruebas de hipótesis se encuentra la prueba de Kruskal-Wallis, la cual es utilizada para determinar si las medianas de dos o más muestras difieren. Esta prueba será utilizada para comprobar si existen diferencias entre los tiempos de ejecución de cada una de las variantes del algoritmo. Para determinar si existen diferencias entre los tiempos de ejecución de las variantes implementadas se definen como hipótesis las siguientes:

H_0 : Los sistemas son iguales en cuanto a tiempo de respuesta.

H_1 : Uno de los sistemas es diferente al otro en cuanto a su tiempo de respuesta.

Donde H_0 es la hipótesis nula y H_1 es la hipótesis alternativa. Se utiliza un α (nivel de significación) = 0,05. En la Fig. 5 se muestra el resultado de realizar la prueba usando el software Minitab 16. Para ello fueron utilizados los tiempos de ejecución que se muestran en la Tabla 1.

Prueba de Kruskal-Wallis: Tiempo vs. Algoritmo

Prueba de Kruskal-Wallis en Tiempo

Algoritmo	N	Mediana	Clasificación del promedio	Z
Paralelo	10	22,33	5,5	-3,78
Secuencial	10	171,04	15,5	3,78
General	20		10,5	

H = 14,29 GL = 1 P = 0,000

Fig5: Prueba Kruskal-Wallis.

Al realizar la prueba se obtuvo un valor $p=0,000$ como resultado de un redondeo. Como este valor es menor que el α se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo que se puede concluir que los dos sistemas difieren en cuanto al tiempo de respuesta. Debido al resultado obtenido en la prueba de Kruskal-Wallis se pasa

demostrar que la versión paralela presenta menor tiempo de respuesta que la versión secuencial del sistema. Para ello se realizará una comparación entre la variante secuencial y la paralela usando la prueba de Mann-Whitney. Esta prueba permite determinar si la mediana de una muestra es diferente que la otra.

Para realizar la prueba de Mann-Whitney mostrada en la Fig. 6, se fija el nivel de significancia $\alpha = 0.05$ y se plantean las siguientes hipótesis:

H_0 : El sistema secuencial es igual que la versión paralela en cuanto a tiempo de respuesta.

H_1 : El sistema secuencial presenta tiempos de respuesta mayores que la versión paralela.

Prueba de Mann-Whitney e IC: Secuencial; Paralelo

	N	Mediana
Secuencial	10	171,04
Paralelo	10	22,33

La estimación del punto para ETA1-ETA2 es 150,61
95,5 El porcentaje IC para ETA1-ETA2 es (94,32;169,21)
W = 155,0
Prueba de ETA1 = ETA2 vs. ETA1 > ETA2 es significativa en 0,0001

Fig6: Prueba Mann-Whitney.

Al realizar la prueba se obtuvo un valor $p=0,0001$, que es menor que el α por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa.

A partir de lo anteriormente planteado se puede concluir que el diseño secuencial del componente gs-opensl presenta tiempos de respuesta mayores que la versión paralela del mismo.

Con el objetivo de demostrar que la variante paralela mejora el tiempo de ejecución de la variante secuencial y establecer comparaciones entre ellas se utilizarán pruebas estadísticas.

Dentro de las pruebas de hipótesis se encuentra la prueba de Kruskal-Wallis, la cual es utilizada para determinar si las medianas de dos o más muestras difieren. Esta prueba será utilizada para comprobar si existen diferencias entre los tiempos de ejecución de cada una de las variantes del algoritmo. Para determinar si existen diferencias entre los tiempos de ejecución de las variantes implementadas se definen como hipótesis las siguientes:

H_0 : Los sistemas son iguales en cuanto a tiempo de respuesta.

H_1 : Uno de los sistemas es diferente al otro en cuanto a su tiempo de respuesta.

Donde H_0 es la hipótesis nula y H_1 es la hipótesis alternativa. Se utiliza un α (nivel de significación) =0,05. En la Fig. 5 se muestra el resultado de realizar la prueba usando el software Minitab 16. Para ello fueron utilizados los tiempos de ejecución que se muestran en la Tabla 1.

Prueba de Kruskal-Wallis: Tiempo vs. Algoritmo

Prueba de Kruskal-Wallis en Tiempo

Algoritmo	N	Mediana	Clasificación del promedio	Z
Paralelo	10	22,33	5,5	-3,78
Secuencial	10	171,04	15,5	3,78
General	20		10,5	

H = 14,29 GL = 1 P = 0,000

Fig5: Prueba Kruskal-Wallis.

Al realizar la prueba se obtuvo un valor $p=0,000$ como resultado de un redondeo. Como este valor es menor que el α se rechaza la hipótesis nula y se acepta la hipótesis alternativa, por lo que se puede concluir que los dos sistemas difieren en cuanto al tiempo de respuesta. Debido al resultado obtenido en la prueba de Kruskal-Wallis se pasa demostrar que la versión paralela presenta menor tiempo de respuesta que la versión secuencial del sistema. Para ello se realizará una comparación entre la variante secuencial y la paralela usando la prueba de Mann-Whitney. Esta prueba permite determinar si la mediana de una muestra es diferente que la otra.

Para realizar la prueba de Mann-Whitney mostrada en la Fig. 6, se fija el nivel de significancia $\alpha = 0.05$ y se plantean las siguientes hipótesis:

H_0 : El sistema secuencial es igual que la versión paralela en cuanto a tiempo de respuesta.

H_1 : El sistema secuencial presenta tiempos de respuesta mayores que la versión paralela.

Prueba de Mann-Whitney e IC: Secuencial; Paralelo

	N	Mediana
Secuencial	10	171,04
Paralelo	10	22,33

La estimación del punto para ETA1-ETA2 es 150,61
 95,5 El porcentaje IC para ETA1-ETA2 es (94,32;169,21)
 W = 155,0
 Prueba de ETA1 = ETA2 vs. ETA1 > ETA2 es significativa en 0,0001

Fig6: Prueba Mann-Whitney.

Al realizar la prueba se obtuvo un valor $p=0,0001$, que es menor que el α por lo que se rechaza la hipótesis nula y se acepta la hipótesis alternativa.

A partir de lo anteriormente planteado se puede concluir que el diseño secuencial del componente gs-opens presenta tiempos de respuesta mayores que la versión paralela del mismo.

Conclusiones

Con el desarrollo del trabajo expuesto se logró cumplir con el objetivo planteado. Se redujo el tiempo de respuesta de los Servicios Basados en la Localización en su forma estandarizada OpenLS cuando se les realiza una petición en lotes, específicamente al servicio de geocodificación de direcciones postales para Cuba.

Como complemento a los antes planteado se puede arribar a las siguientes conclusiones:

- El uso de técnicas de computación paralela y distribuida en un sistema informático puede reducir en gran medida los tiempos de respuestas de los mismos.
- Los componentes desarrollados para GeoServer pueden hacer uso de la computación paralela utilizando las técnicas de memoria compartida brindadas por el lenguaje de programación Java 8.
- Un elemento de vital importancia en la comparación de rendimiento de dos sistemas es el análisis experimental que se le realice a los datos obtenidos con cada versión del sistema.

Se logró disminuir grandemente los tiempos de respuesta del componente gs-opensl utilizando en su diseño e implementación técnicas de computación paralela y distribuida. Con el desarrollo del trabajo expuesto se logró cumplir con el objetivo planteado. Se redujo el tiempo de respuesta de los Servicios Basados en la Localización en su forma estandarizada OpenLS cuando se les realiza una petición en lotes, específicamente al servicio de geocodificación de direcciones postales para Cuba.

Como complemento a los antes planteado se puede arribar a las siguientes conclusiones:

- El uso de técnicas de computación paralela y distribuida en un sistema informático puede reducir en gran medida los tiempos de respuestas de los mismos.
- Los componentes desarrollados para GeoServer pueden hacer uso de la computación paralela utilizando las técnicas de memoria compartida brindadas por el lenguaje de programación Java 8.
- Un elemento de vital importancia en la comparación de rendimiento de dos sistemas es el análisis experimental que se le realice a los datos obtenidos con cada versión del sistema.

Se logró disminuir grandemente los tiempos de respuesta del componente gs-opensl utilizando en su diseño e implementación técnicas de computación paralela y distribuida.

Referencias

- [1] E. S. Ansola, "Servicios Basados en Localización para una Infraestructura de Datos Espaciales," Tesis en opción al título de máster en Informática Aplicada, Ingeniería Informática, Instituto Superior "José Antonio Echeverría" (ISPJAE), Habana, Cuba, 2005.
- [2] K.-t. Chang, *Introduction to Geographic Information Systems*. Boston: McGraw-Hill Higher Education 2006.

- [3] GSDI, *Developing Spatial Data Infrastructures: The SDI Cookbook* vol. 2: GSDI, 2004.
- [4] Francisco de Asís López Fuentes, *Sistemas distribuidos*. México: Universidad Autónoma Metropolitana Unidad Cuajimalpa, 2015.
- [5] Open Geospatial Consortium Inc., *OpenGIS Web Feature Service 2.0 Interface Standard*, 2014.
- [6] I. Open Geospatial Consortium, *OpenGIS® Web Map Server Implementation Specification* vol. 1.3.0, 2006.
- [7] Open Geospatial Consortium Inc., *OpenGIS® Catalogue Services Specification*, 2007.
- [8] Open Geospatial Consortium, "OpenGIS Location Services (OpenLS): Core Services," E. Marwa Mabrouk, Ed., ed: OGC, 2008.
- [9] Leandry Girón Lima, "Servicios Basados en la Localización para GeoServer," Trabajo de diploma para optar por el título de Ingeniería Informática, Facultad de Ingeniería Informática, Universidad Tecnológica de la Habana "José Antonio Echeverría", La Habana, 2017.
- [10] GeoServer, "GeoServer User Manual Release," ed, 2014.
- [11] Robert Liguori and Patricia Liguori, *Java 8 Pocket Guide*. United States of America.: O'Reilly Media, Inc., 2014.