

Módulo de comunicaciones, en tiempo real, para Sistemas de Control de Flotas basados en la Web.

Ing. Liester Cruz Castro.

Especialista en bases de datos espaciales y profesor de la Facultad 6, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.
lcruz@uci.cu

Ing. Miosotis Aida Troche Rodríguez.

Especialista en Gestión de Configuración, Facultad 6, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.
miosotisaida@uci.cu

Ing. Javier Rodríguez Ramírez.

Profesor del Instituto Superior Politécnico "José A. Echeverría", Calle 114 entre 119 y 129, Marianao, La Habana, Cuba.
jramirez@cemat.cujae.edu.cu

Ing. Arianna Guerra Medrano.

Especialista en la Empresa Cubana de Software para la Salud, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.
ary_medrano@softel.sld.cu

Resumen

El Protocolo de Transferencia de Hipertexto (HTTP) es ampliamente utilizado. Sin embargo, por sus características de funcionamiento, resulta insuficiente para nuevos sistemas informáticos que requieren implementación en tiempo real, siendo el caso de los Sistemas de Control de Flotas (SCF). Por tal motivo es necesario establecer un protocolo de comunicación más rápido para este tipo de sistemas. Para desarrollos donde la velocidad y eficiencia de las comunicaciones juegan un papel fundamental es necesario tener en cuenta la estructura y formas sobre la cual fluye la información, ya que de esta depende el éxito de los resultados a obtener. Por tal motivo se tuvo como objetivo el desarrollo de un módulo de comunicaciones, en tiempo real, para SCF; haciéndose uso de metodologías, protocolos y tecnologías existentes. El módulo de comunicaciones obtenido con la ayuda del protocolo Websockets mejora la toma de decisiones sobre SCF, evidenciándose esto en las pruebas realizadas.

Palabras claves: rapidez; comunicaciones; control; flotas; módulo.

Introducción

Las Aplicaciones Ricas en Internet (RIA) son un nuevo tipo de aplicaciones que intentan unir, en el mismo entorno, las potencialidades de las aplicaciones de escritorio con las facilidades de los sistemas Web. Entre las RIA más difundidas se encuentran los Sistemas de Información Geográfica (SIG). Un SIG es *“una integración organizada de hardware, software y datos geográficos diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión”* (Santovenia 2009). Numerosos son los ejemplos de SIG, que interaccionan con dispositivos GPS

(Sistema de Posicionamiento Global) y le brindan a los usuarios nuevos valores añadidos, los cuales influyen directamente en la toma de decisiones. Esta nueva unión permitió que surgieran los Sistemas de Control de Flotas, que permiten entre otras tareas garantizar el control y seguimiento de objetos móviles que proveen su posición, así como otros datos como son la velocidad, aceleración y la altura.

Generalmente los SCF Web a nivel mundial están basados en el protocolo HTTP, debido a su amplia utilización en internet. Estos sistemas como principal funcionalidad tienen la de garantizar una correcta y eficiente toma de decisiones; aspectos como la capacidad de procesamiento y rapidez en el flujo de comunicación son necesarios para ello, en estas situaciones la comunicación a través del protocolo HTTP no es eficiente. Se necesita entonces contar con un protocolo que garantice un medio de comunicación rápido, que le dé al usuario la sensación de tiempo real en las operaciones que realiza dentro de la flota que controla. Una vez identificado este protocolo, sería necesario integrarlo con las funcionalidades básicas de un SCF Web en un canal de comunicación donde esté presente. Para todo esto, se necesita contar con un diseño estructural donde se especifique cada una de las partes de este nuevo sistema a desarrollar, la relación que existe entre cada una de ellas y que parte va a fungir como entidad comunicadora, siendo esta el centro de la estructura. Todo lo anteriormente explicado condujo a que se tuviera como objetivo el desarrollo de un módulo de comunicaciones, en tiempo real, para SCF basados en la Web. El presente trabajo describe como se le dio solución a la problemática planteada para cumplimentar el objetivo trazado.

Desarrollo

Materiales y Métodos

Para lograr un correcto diseño arquitectónico se hizo uso de los métodos científicos analítico-sintético y la observación permitiendo identificar las herramientas, tecnologías y metodología de desarrollo más adecuadas. La metodología de desarrollo utilizada fue RUP (Proceso Unificado de Rational), ya que esta permite generar documentación y vistas arquitectónicas que pueden servir de guía para futuras modificaciones. Para complementar la metodología utilizada, se hizo uso del Lenguaje de Modelado Unificado (UML) que *es actualmente uno de los lenguajes de modelado más usado a nivel mundial por las grandes empresas productoras de software, ya que permite visualizar, especificar, construir y documentar los artefactos de la aplicación de forma eficiente y entendible* (OMG 2003). La herramienta CASE (Ingeniería de Software Asistida por Ordenador) utilizada para la modelación de los diagramas con UML fue Visual Paradigm, haciendo uso de una licencia permisiva. Para desarrollar el módulo se usó el Entorno de Desarrollo Integrado (IDE) NetBeans en su versión 7.3.

Para el proceso de comunicación entre los componentes del SCF, se define utilizar el protocolo WebSocket debido a las ventajas que este medio de comunicación provee para aplicaciones en tiempo real. *El objetivo de esta tecnología es proveer un mecanismo para aplicaciones basadas en navegadores que necesiten una comunicación de doble sentido con servidores y que no requieran abrir múltiples conexiones HTTP* (Fette, 2011). Las principales ventajas de este protocolo con respecto a HTTP, es que los mensajes pueden viajar en ambos sentidos a la misma vez, y la minimalidad de los encabezados. En el protocolo WebSocket los mensajes tienen como encabezado entre 4 y 14 bytes, mientras que en HTTP oscilan aproximadamente entre 200 bytes y 2 Kb. El uso de este protocolo permite la eliminación de la transmisión de información superflua por lo cual el tráfico de red es mucho menor. Se define el uso de Node.js

(Servidor Web) como tecnología para el desarrollo debido a los altos rendimientos que puede alcanzar en cuanto a consumo de recursos del servidor.

Resultados y Discusión

El módulo informático obtenido sirve de intermediario en las comunicaciones entre los módulos de SCF de forma tal que, aprovechando las ventajas de la tecnología WebSocket, las velocidades de interacción entre dichos módulos sean lo suficientemente rápidas como para considerar a la aplicación un sistema en tiempo real. Se definió un formato de comunicación entre los módulos de la aplicación que contiene la información necesaria para que el módulo de comunicaciones, actuando de intermediario, re-direccione las peticiones de un módulo origen a uno destino. El campo Sec-WebSocket-Protocol de WebSocket, será utilizado en beneficio de las necesidades de los SCF, para indicar el tipo de protocolo a nivel de aplicación que será utilizado para darle formato a la información a transmitir o que se desea recibir. De esta manera, los mensajes transmitidos primeramente irán sobre el protocolo TCP, encapsulado dentro de este tendrán el formato WebSocket, y finalmente, dentro de WebSocket tendrán el formato definido para la aplicación (PA o protocolo de aplicación) en el cual viajarán la información enviada por algún módulo cliente o algún sub-protocolo de aplicación con datos contenidos.

Se obtuvo en la implementación código fuente que puede ejecutarse en múltiples plataformas, debido a la utilización del lenguaje JavaScript tanto de lado cliente como del lado servidor (utilizado en Node.js). Todo lo implementado puede reutilizarse con facilidad y responde a las políticas de desarrollo de software libre, potenciando esto a obtener mejoras del módulo por parte de las comunidades de desarrolladores. Dicho módulo una vez ejecutado funciona como un proceso que es capaz de garantizar de forma instantánea la comunicación entre módulos clientes y servidores. El proceso de comunicación en SCF comprende las funcionalidades siguientes: establecer conexión, enviar solicitud de acción, re-direccionar petición de cliente y validar petición.

Una vez finalizado el desarrollo e integrado el resultado en un SCF se procedió a realizarle un conjunto de pruebas con el objetivo de conocer si se cumplieron los objetivos propuestos. Se le realizaron al módulo de comunicaciones, pruebas de estrés y pruebas de sistema. Las pruebas de estrés estuvieron dirigidas a encontrar el volumen de datos o de tiempo en que el módulo comenzó a fallar o fue incapaz de responder a las peticiones, y la capacidad del módulo de aceptar nuevas conexiones. Las pruebas del sistema se realizaron sobre el módulo en su conjunto, así como su relación con otros subsistemas.

La computadora personal (PC) que se utilizó tenía como características en su hardware: 1GB de memoria RAM, microprocesador Core 2 Duo de 2.20GHz y sistema operativo Ubuntu 11.10. Para hacer las pruebas de carga se utilizaron dos de las principales opciones de la herramienta ApacheBench, una es la cantidad de peticiones enviadas al servidor y la otra es la cantidad de peticiones concurrentes. Las pruebas sobre el módulo fueron realizadas también en tecnologías clásicas que son utilizadas en numerosas aplicaciones Web, tal es el caso del servidor Apache que permite la interacción con aplicaciones desarrolladas con el lenguaje PHP entre otros. La respuesta que dieron ambos servidores ante cada petición fue un simple "Hola Mundo", los resultados se reflejan en la siguiente tabla.

Tecnología	Peticiones por segundos	Tiempo promedio de respuesta (milisegundos)
------------	-------------------------	---

Caso A: Para 10000 peticiones y 100 concurrencias		
Apache-PHP	5099.02	19.612
Módulo-Node.js	6190.00	16.155
Caso B: Para 10000 peticiones y 500 concurrencias		
Apache-PHP	220.64	2266.134
Módulo-Node.js	6186.40	80.822
Caso C: Para 100000 peticiones y 1000 concurrencias		
Apache-PHP	La PC falló	La PC falló
Módulo-Node.js	6124.94	163.267

Tabla 1. Prueba de Carga a Node.js. Casos A, B y C.

En la prueba del caso B, Apache-PHP consumió casi todos los recursos de la PC mientras Node.js se mantuvo estable. En la prueba del Caso C se intentó llevar al límite a ambas tecnologías. Apache con cada petición crea un nuevo hilo, al tener varias peticiones concurrentes la memoria usada es de aproximadamente 4MB por la cantidad de peticiones concurrentes. Es la razón por la que en la última prueba la computadora falló. Sin embargo, el módulo de comunicaciones consumió aproximadamente de 20MB a 27MB en todas las pruebas. Esto demuestra que los resultados medidos en el producto propuesto proporcionan ventajas sobre tecnologías que utilizan HTTP que son ampliamente utilizadas en el mundo.

Se realizó una prueba para medir la capacidad del módulo de comunicaciones para aceptar conexiones. En una computadora con Sistema Operativo Windows 7, procesador Intel Centrino Core 2 Duo a 2.1 GHz y 3.00 GB de memoria RAM; el servidor fue capaz de aceptar 4382 conexiones concurrentes. En esta prueba queda en evidencia la capacidad de soportar un gran número de conexiones sin afectar notablemente el rendimiento.

```

Clients connected: 4380
.....
New socket connected:
< address: '127.0.0.1', family: 'IPv4', port: 8000 >
.....
Clients connected: 4381
.....
New socket connected:
< address: '127.0.0.1', family: 'IPv4', port: 8000 >
.....
Clients connected: 4382
.....
New socket connected:
< address: '127.0.0.1', family: 'IPv4', port: 8000 >
.....

```

Figura 1. Total de clientes conectados.

El objetivo de la prueba del sistema fue demostrar el correcto funcionamiento del módulo de comunicaciones al re-direccionar una solicitud de acción desde un módulo suscriptor hacia un módulo receptor y obtener eficientemente el resultado de esta orden en el suscriptor. Para esto se envió desde el modulo suscriptor la orden de sumar dos números, el módulo receptor recibió los datos y realizó la suma, el resultado se retornó al origen; con esto se comprobó que el sistema funcionaba correctamente. Conjunto con la comprobación del funcionamiento del módulo de comunicaciones, se midió el tiempo en que se obtuvo la respuesta en el módulo suscriptor, aspecto que ejempli-

fica una vez más la eficiencia en la comunicación que brinda el producto obtenido, reflejada en la rapidez con que se completó el proceso requerido. La siguiente figura muestra el tiempo que demoró la transacción.

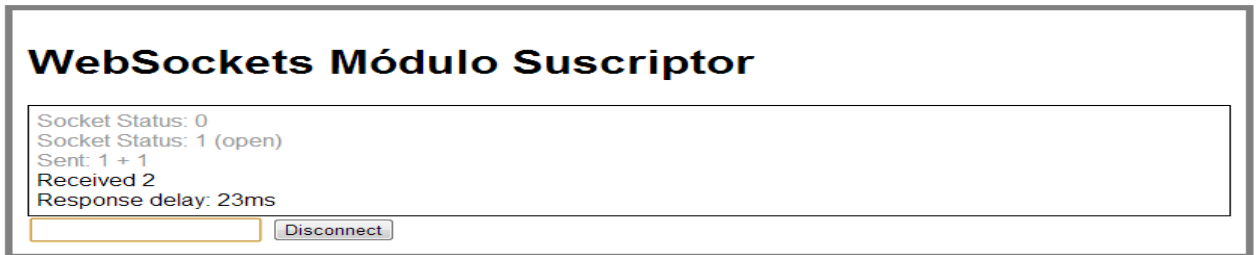


Figura 2. Interfaz del Módulo Suscriptor.

Conclusiones

El surgimiento de los SCF deviene de garantizar la toma de decisiones sobre los objetos móviles asociados, para ello es necesaria su visualización en tiempo real. Siguiendo esta premisa se realizó la elección de un protocolo de comunicación que simula la realización en tiempo real de las actividades de un SCF. Es por este motivo que el módulo propuesto genera un cambio en cuanto a la realización de los SCF que funcionan con HTTP. Por otra parte las pruebas realizadas sobre los resultados obtenidos manifiestan los beneficios de utilizar este módulo, no solo en comunicaciones en SCF sino también en otros tipos de sistemas, que necesiten mecanismos de comunicación similares.

Referencias

- SANTOVENIA, J., TARRAGÓ, C. Y CAÑEDO, R. "Sistemas de información geográfica para la gestión de la información", *ACIMED*, 20(5): p- 72-75, 2009.
- OBJECT MANAGEMENT GROUP INC. "Unified Modelling Language Specification". Massachusetts: OMG, v1.5, 2003.
- FETTE, I., MELKINOV, A. The WebSocket Protocol. [En línea] 2011. [Citado el: 27 de febrero de 2014.] <https://tools.ietf.org/html/rfc6455>.



www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x