

Optimización del desempeño de un Sistema Web para Modelos de Inferencia Difusa mediante Técnicas no invasivas

Miguel Ángel Uc Dzib¹ Victor Hugo Menéndez Domínguez² Salvador Medina Peralta³

¹ Universidad Autónoma de Yucatán, Facultad de Matemáticas,
Anillo Periférico Norte, Chuburná Hidalgo Inn, 13615 Mérida, Yucatán.
mgelster@gmail.com

² Universidad Autónoma de Yucatán, Facultad de Matemáticas,
Anillo Periférico Norte, Chuburná Hidalgo Inn, 13615 Mérida, Yucatán.
mdoming@uady.mx

³ Universidad Autónoma de Yucatán, Facultad de Matemáticas,
Anillo Periférico Norte, Chuburná Hidalgo Inn, 13615 Mérida, Yucatán.
mperalta@uady.mx

Fecha de recepción: 6 de noviembre 2015

Fecha de aceptación: 16 de diciembre 2015

Resumen. Se evaluó el rendimiento de un Sistema Web que permite simular modelos de inferencia difusa (SWID), se presenta una propuesta genérica para optimizar la ejecución de un SWID en un servidor Web (Apache), utilizando una configuración no invasiva que hace uso de la compresión y el caché del servidor Web. La reciente necesidad de utilizar modelos matemáticos en disciplinas como la Biología, Medicina, Psicología, Educación, entre otras, ha estimulado el interés en el empleo de nuevas técnicas para su generación. Los sistemas de inferencia difusa permiten modelar procesos complejos o sistemas incompletos o inciertos, se utilizó la prueba de wilcoxon y t – student para muestras pareadas para ver diferencias con la optimización y sin la optimización.

Palabras Clave: SWID, Rendimiento, wilcoxon, t - student.

1 Introducción

La accesibilidad de una aplicación se refiere a su utilización, en forma satisfactoria, por un mayor número de personas, sin importar sus limitaciones intrínsecas o derivadas del entorno en que se encuentran [9]. El desempeño de una aplicación, entendiéndose su optimización, repercute en gran medida a la accesibilidad. Se hace especial énfasis en limitar el uso de imágenes, así como reducir su resolución y tamaño con el propósito de disminuir el tiempo de espera con respecto a la descarga de la página. Internet ha tenido una marcada evolución en los últimos años, permitiendo el acceso y el intercambio de información de manera ágil, con características de flexibilidad en la tecnología de acceso y con capacidades de integración a nivel de servicios [1]. Se propone el empleo de técnicas para optimizar la descarga y despliegue de las páginas. Por esta razón se evalúa un sistema web para simular modelos de inferencia difusa (SWID) en áreas como la Medicina, Biología, Psicología entre otras.

Hemos seleccionado a HTML, PHP y AJAX como lenguajes de programación para construir "SWID", utiliza el lenguaje de programación R[4] como motor de inferencia estadístico, SWID es un sistema Web que permite generar modelos de inferencia difusa en áreas como la Medicina, Biología, Psicología. Se basa en PHP-AJAX que está dirigido por el servidor Apache en cualquier sistema operativo como Linux, Windows y MAC. SWID ofrece las funciones de gestión de datos, funciones de pertenencia y análisis de datos para 2 tipos de algoritmos, el primer algoritmo implementa las reglas de clasificación fuzzy usando la técnica Chi's (FRBCS.CHI) [5] y el segundo utiliza las reglas de clasificación con un factor de peso (FRBCS.W) [6]. Hay dos características particulares en SWID: (1) Genera predicciones de sistemas de inferencia difuso en tiempos óptimos y (2) los gráficos de función de pertenencia para un determinado número de etiquetas.

2 Estado del arte

El objetivo de este apartado es proporcionar los antecedentes relacionados con la optimización de del desempeño de un sistema web y los sistemas de inferencia difusa.

En el desarrollo de aplicaciones Web se han establecido colecciones de directrices [10] que generalmente se enfocan en controlar la estructura de los contenidos para una fácil lectura y comprensión por los usuarios. Dos técnicas muy socorridas para mejorar el desempeño de una aplicación Web son la compresión de información y

el uso de caché en el cliente y el servidor Web [10]. El diseño de la arquitectura de software es un pilar importante en el desarrollo de software, ya que la manera en que se estructura un sistema tiene impacto directo sobre la capacidad de este para satisfacer los atributos de calidad del sistema [2]. Un Sistema de inferencia difusa emplea reglas difusas “si-entonces” que pueden modelar los aspectos cualitativos del conocimiento humano y procesos de razonamiento sin emplear análisis cuantitativos precisos. Esta modelación difusa, tiene impacto práctico en el control, la predicción y la deducción [3].

La lógica difusa es una extensión de la lógica matemática basada en la teoría de conjuntos difusos y sistemas lógicos de varios valores infinitos [11]. La lógica difusa se ha convertido en un campo de la investigación y el desarrollo importante en muchas disciplinas como las matemáticas, la lógica, la inteligencia artificial y la filosofía desde 1965 [11].

La base de conocimientos de un controlador difuso consiste en una colección de reglas que describen las acciones de control. El rendimiento del control difuso depende en gran medida de si las normas de control son razonables o no [12].

Hay diferentes modos para derivar de ellos [12]:

- Basado en la experiencia de expertos y Control de Ingeniería del Conocimiento.
- Sobre la base de las acciones de control del operador.
- Basado en el modelo difuso de un Proceso.
- Sobre la base de aprendizaje o de organización Autocontrol.

Las posibilidades de aplicación de los algoritmos evolutivos en el campo de la lógica difusa esta documentada por una serie de publicaciones recientes de investigación , que a grandes rasgos permite optimizar las funciones de pertenencia de los conjuntos borrosos, además de optimizar el aprendizaje automático de las reglas difusas [13].

Las funciones de pertenencia Fuzzy proporcionan la caracterización de conjuntos difusos mediante el establecimiento de un conexión entre términos lingüísticos (como " lento " , "medio " , "rápida" para una variable de velocidad) y los valores numéricos precisos de variables en un sistema físico. Una función de pertenencia difusa se aproxima a la confianza con la que un valor numérico es descrito por un término lingüístico, la correcta elección de las funciones de pertenencia , sin embargo , no es en absoluto triviales pero juega un papel crucial en el éxito de una aplicación [13].

El reciente desarrollo de técnicas de detección de partículas moleculares en células vivas ha estimulado el interés en el desarrollo de las nuevas técnicas de gran alcance para rastrear las partículas moleculares en las células vivas. Un tipo especial de imágenes de microscopía celulares se trata de la formación y transporte de fosas y vesículas revestidas de clatrina. Por lo tanto, un algoritmo de correspondencia de movimiento basado en el sistema basado en reglas fuzzy permite resolver el problema de la asociación ambigua encontrado en estas imágenes dinámicas, en vivo de células de conjuntos de clatrina. Los resultados muestran que este método puede rastrear con precisión la mayoría de las partículas en las imágenes de alto volumen[14].

3 Metodología

Todos los experimentos se ejecutaron en una computadora Toshiba satélite con procesador AMD A6, 4 GB de RAM y sistema operativo Debian 8.

Para activar el control de caché del cliente, se han incorporado las siguientes directivas de configuración del servidor Apache:

```
<IfModule mod_expires.c>
ExpiresDefault"access plus 1 month"           (2)
FileETag None
</IfModule>
```

Para activar la compresión se utilizó el módulo Apache mod_deflate que permite controlar el grado de compresión y su velocidad para no afectar el desempeño del servidor, las directivas utilizadas son:

```
<IfModule mod_deflate.c>
AddOutputFilterByType DEFLATE text/html text/css
application/javascript
BrowserMatch ^Mozilla/4 gzip-only-text/html   (3)
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
```

</IfModule>

Para validar la configuración aplicada al servidor Web se ejecutaron 100,150, 200, 250, 300, 350, 400, 450, 500, 550 y 600 peticiones de manera aleatoria con un nivel de concurrencia de 10 utilizando la herramienta ab (Apache HTTP server benchmarking tool.) [7].

Las gráficas estadísticas y las pruebas de hipótesis se obtuvieron con el programa estadístico R 3.0.1 [4], los resultados de las pruebas estadísticas fueron considerados significativos si $P < 0.05$.

Debido al incumplimiento del supuesto de normalidad para la diferencia de peticiones por segundo ($SW = 0.8473$, $P = 0.03939$) se utilizó la prueba de rangos con signo de Wilcoxon para muestras relacionadas [8]. Con respecto a la razón de transferencia el supuesto de normalidad se cumplió ($SW = 0.9222$, $P = 0.3371$) por lo que se utilizó la prueba T – Student para muestras pareadas [8].

4 Resultados experimentales

Los resultados obtenidos al simular las peticiones por algoritmo se presenta en las figura 1.

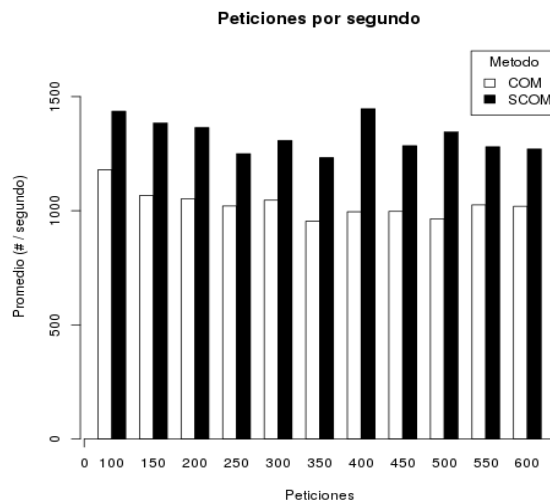


Figura 1. Número de peticiones contra el tiempo promedio utilizando compresión (COM) y sin compresión (SCOM).

La figura 1 muestra el número de peticiones (100,150, 200, 250, 300, 350, 400, 450, 500,550 y 600) y el tiempo promedio por segundo, se puede ver que el promedio cuando se aplica la compresión es menor ($W = 66$, $P = 0.0004883$).

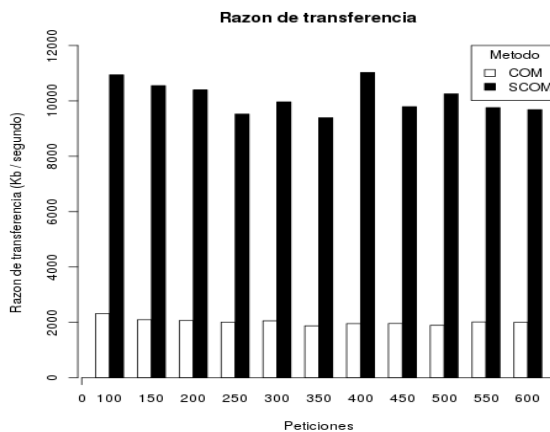


Figura 2. Número de peticiones contra razón de transferencia utilizando compresión (COM) y sin compresión (SCOM).

La figura 2 muestra la razón de transferencia por petición (100,150, 200, 250, 300, 350, 400, 450, 500,550 y 600) puede ver que la razón de transferencia cuando se aplica la compresión es menor ($t = 53.1698$, $g1 = 10$, $P = 6.705e-14$).

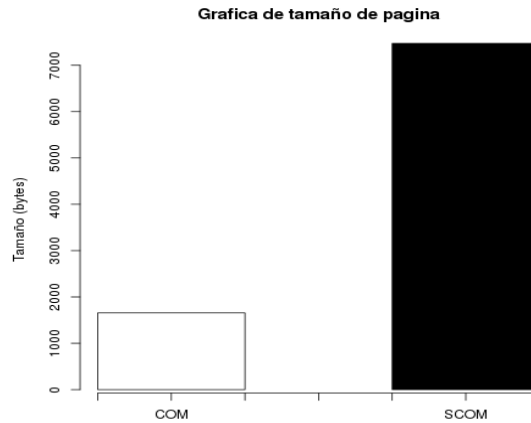
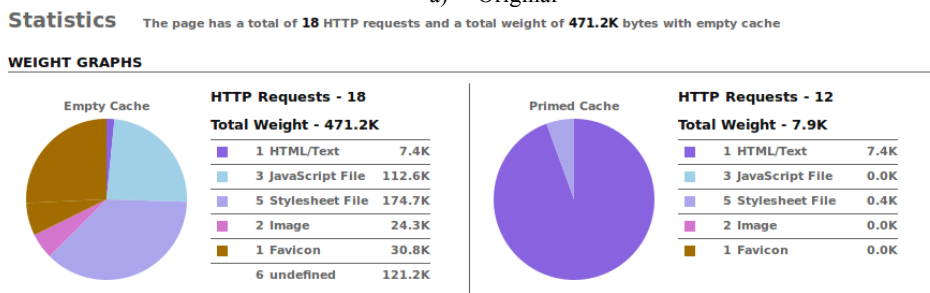


Figura 3. Gráfica del experimento cuando se aplica la compresión y sin compresión.

La figura 3 evalúa el tamaño de la página considerando todos los archivos (HTML, Java Script,css, imágenes, etc.) utilizando la compresión y sin la compresión la cual resultado de 1657 bytes y de 7471 respectivamente, lo cual significa que hay una reducción del 88 % aproximadamente.

a) Original



b) Optimizado

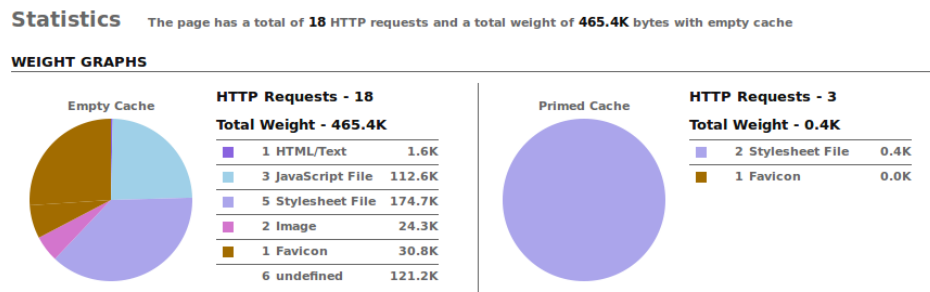


Figura 4. Estadísticas de descarga de la página principal de SWID utilizando la configuración original (a) y la optimizada (b) con la herramienta FireBug.

Figura 4 presenta gráficos de pastel que resaltan el beneficio de utilizar el caché y la compresión, en relación al volumen de datos que es transferido la primera vez que se accede a la página (471.2 Kb) sin compresión contra una visualización posterior de la misma página utilizando la configuración optimizada. En este último caso, el volumen de transferencia se reduce hasta ser 91.5% (0.4 Kb), que es una diferencia muy significativa en términos de percepción del usuario (la página se presentará más rápido). Es importante resaltar que el tamaño real de la página Web utilizada en el experimento es de 471.2 Kb contra los 465.4 Kb que tiene luego de compactarse, una disminución de poco más del 1.23 %.

5 Conclusiones y trabajos futuros de investigación

En este trabajo se ha presentado una propuesta no invasiva para mejorar el desempeño de un Sistema SWID en un servidor Web, en términos de sus tiempos de respuesta. Los resultados resaltan la importancia de una adecuada configuración del servidor Web para garantizar una mejor y más rápida transferencia de información. El uso del servidor Web Apache y la herramienta ab facilitó en gran medida la aplicación de las técnicas, esto gracias a su arquitectura modular que permite incorporar nuevas características, como la compresión, el control de encabezados y otros elementos que favorecen su optimización. Muchas bases de datos y herramientas de análisis de datos se han creado para la investigación bioinformática, generalmente se combinan para ser utilizados en las tareas de recuperación de datos y de análisis de datos por algunos biólogos y científicos en bioinformática, SWID puede ser usado como herramienta de investigación no solo en el área de la bioinformática sino también en otras disciplinas como Medicina, Psicología, Educación, entre otras. Como trabajo futuro se propone evaluar el rendimiento de SWID variando el número de etiquetas y el tipo de función de pertenencia e.g la gaussiana, triangular, así como proponer otros tipos de algoritmos en el área de la lógica difusa.

Referencias

- [1] P. Lago, R. Farenhorst, P. Aygeriou, R.C. de Boer, V. Clerc, A. Jansen, and H. Van Vliet. "The GRIFFIN Collaborative Virtual Community for Architectural Knowledge Management". in Collaborative Software Engineering, Eds. Springer , 2010, pp. 195-217.
- [2] H. Cernvantes. "Arquitectura de Software". SG software guru, México, 2010, Vol. 1. 1870 ----1888.
- [3] Takagi, Tomohiro Sugeno, Michio. "Fuzzy identification of systems and its applications to modeling and control". IEEE Transactions on Systems, Man and Cybernetics, 116-132, 1985.
- [4] Ihaka, Ross. "R : A language for data analysis and graphics". J. Comput. Graph. Stat. , 299- 314, 1996.
- [5] Z. Chi, H. Yan, T. Pham. "Fuzzy algorithms with applications to image processing and pattern recognition". World Scientific, Singapore (1996).
- [6] H. Ishibuchi and T. Nakashima. "Effect of rule weights in fuzzy rule-based classification systems". IEEE Transactions on Fuzzy Systems, vol. 1, pp. 59 - 64 (2001).
- [7] Apache Software Foundation. "Apache HTTP Server Reference Manual - for Apache version 2.2.17". Network Theory Ltd., EUA, 2010.
- [8] Zar, J. H. 1999. "Biostatistical Analysis". 4ª Edición. Prentice Hall. New Jersey, USA. 906 p.
- [9] W3C WAI. "Web Accessibility Initiative (WAI)", W3C (en línea), 2010, consultada por Internet el 18 de mayo de 2013. Dirección de Internet: <http://www.w3.org/WAI/>.
- [10] Souders, S. "High Performance Web Sites: Essential Knowledge for Front-End Engineers". O'Reilly Media, Ed. 1, EUA, 2007.
- [11] Zhaohao Sun, Gavin Finnie. "A Fuzzy Logic Approach to Experience Based Reasoning". International Journal of Intelligent Systems, Vol. 22, 867-889 , 2007.
- [12] Francisco Herrera, Manuel Lozano, Jose Luis Verdegay. "Generating Fuzzy Rules From Examples Using Genetic Algorithms". World Scientific Publishing Co. Pte. Ltd., 1995.
- [13] Thomas Back, Frank Kursawe, "Evolutionary Algorithms for Fuzzy Logic: A Brief Overview". World Scientific Publishing Co. Pte. Ltd., 1995.
- [14] Shan Jiang, Xiaobo Zhou, Tom Kirchhausen, Stephen T. C. Wong. "Tracking Molecular Particles in Live Cells Using Fuzzy Rule-Based System". International Society for Analytical Cytology, 123-139, 2007.