Web Oculta del Lado Cliente: Escala de Crawling

Manuel Álvarez, Fidel Cacheda, Rafael López-García, Víctor M. Prieto
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidade da Coruña
Campus de Elviña s/n, A Coruña, España
mad@udc.es, fidel@udc.es, rafael.lopez@udc.es, victor.prieto@udc.es

***Resumen*—El objetivo de este estudio consiste en la definición de una escala para la clasificación de los sistemas de crawling en base a su efectividad accediendo a la Web Oculta del "lado cliente". Para ello se realiza un análisis exhaustivo de las diferentes tecnologías de lado cliente usadas en las páginas Web 2.0 para crear una escala con distintos niveles de dificultad. Para realizar la clasificación de los diferentes sistemas de crawling en base a la escala definida, se ha creado un sitio web contra el que comprobar su efectividad. También se proponen diferentes métodos de evaluación de la efectividad de los crawlers en base a la escala. Para la realización del estudio se han considerado tanto los crawlers de los principales buscadores web como otros sistemas de crawling OpenSource y comerciales.**

***Palabras Clave*—Web Search, crawler, Web Oculta, Web Spam, JavaScript, Redirection Spam**

I. Introducción

La WWW constituye actualmente el mayor repositorio de información jamás construido. Pero tan importante como almacenar gran cantidad de información es la gestión de la misma para permitir localizar, acceder y recopilar la que satisface las necesidades de un usuario. Los sistemas que permiten esta tarea son los crawlers, programas capaces de procesar y analizar la Web. Se puede decir que un crawler recorre los diferentes URL descubiertos, en un cierto orden, analiza el contenido descubierto y lo procesa para obtener nuevos URL que serán tratados. Existen diferentes tipos de sistemas de crawling en función de su ámbito: globales, orientados a recuperar toda o gran parte de la información de la Web, o dirigidos, orientados a una parte concreta, más reducida, de la Web.

Desde sus orígenes los sistemas de crawling han tenido que enfrentarse a sitios web orientados a usuarios humanos: navegaciones a través de menús emergentes, diferentes capas de datos que se ocultan o hacen visibles dependiendo de las acciones del usuario, sistemas de redirecciones o mecanismos de mantenimiento de sesión. El conjunto de tecnologías que posibilitan la inclusión de los aspectos comentados en los sistemas de crawling representa un gran desafío, debido a que obligan a implementar técnicas para tratarlas de forma adecuada. Los sitios web que utilizan estas tecnologías forman la que se conoce como Web Oculta [1], por contener información que no es alcanzable por la mayor parte de los sistemas de crawling. A su vez, ésta se puede dividir en Web Oculta del lado cliente o del lado servidor. En este artículo se analiza cómo tratan las tecnologías del lado cliente (Web Oculta del lado cliente) los crawlers, para intentar determinar si el uso de dichas tecnologías afecta a la visibilidad de aquellas páginas o sitios Web que las usen. Se ha realizado un análisis de las tecnologías del lado cliente más utilizadas en la creación de páginas web, como son JavaScript [2], AJAX (Asynchronous, JavaScript, And XML) [3], VbScript [4] y Flash [5]. También se han analizado problemáticas como Redirection Spam [6] y Cloacking [7] para intentar detectar la utilización de estas tecnologías para fines ilícitos.

Una vez analizadas las diferentes tecnologías, se ha realizado una enumeración de las dificultades que se le pueden presentar a los crawlers durante su recorrido, generando una escala con diferentes niveles. Para clasificar los sistemas de crawling en base a dicha escala, se ha construido un sitio Web que genera de forma dinámica enlaces, según las dificultades propuestas. Se han obtenido resultados tanto para los crawlers de los principales buscadores, como para los crawlers OpenSource [8] [9] y aquellos con licencia comercial.

La estructura del artículo es la siguiente. En la sección II se comentan los trabajos relacionados, los sistemas de crawling de la Web oculta del lado cliente y la problemática del Web Spam. La sección III introduce las tecnologías de lado cliente y su uso para la construcción de sitios web. La sección IV comenta las ocurrencias más habituales de las tecnologías explicadas para la generación de sitios web. A partir de dichas ocurrencias se crean una serie de niveles de dificultad que deberían de ser tratados por un crawler que pretenda obtener toda la información de la Web Oculta del lado cliente. A continuación se define la escala propuesta para la clasificación de los crawlers, junto con cuatro métodos que permiten evaluarlos respecto a dicha escala. La sección V describe el sitio web creado para la realización de experimentos. En la sección VI se discuten los resultados obtenidos para los diferentes crawlers y por último en las secciones VII y VIII se comentan las conclusiones obtenidas y posibles trabajos futuros.

II. Trabajos relacionados

Son muchos los estudios relacionados con el tamaño de la Web y con la caracterización de su contenido. Sin embargo, menos han sido los que se han ocupado de clasificarla en base a la dificultad que presenta hacia los crawlers. Según los datos presentados en [10] y [11] actualmente el 90% de las páginas web usan JavaScript. En 2006, M. Weideman y F. Schwenke [12] publicaron un estudio que analizaba la importancia del uso de JavaScript en la visibilidad de un sitio Web, concluyendo que la mayor parte de los crawlers no lo tratan.

Desde el punto de vista de los sistemas de crawling, son numerosos los trabajos orientados a crear sistemas que sean capaces de tratar la Web Oculta. Los crawlers de la Web Oculta del lado servidor se ocupan de la amplia cantidad de sitios Web en los cuales se accede al contenido mediante

formularios. Este tipo de contenido es de gran cantidad y calidad. Existen investigaciones que abordan los retos marcados por la Web Oculta del lado servidor, destacando HiWE [13] por ser uno de los sistemas pioneros. También Google [14] presentó las técnicas que utiliza para el acceso a información a través de formularios. Respecto a la Web Oculta de lado cliente [15], son menos los estudios y básicamente se resumen en las siguientes dos aproximaciones: acceder al contenido y enlaces mediante intérpretes que permitan ejecutar los scripts [16] [17], o bien utilizar mini-navegadores como en el sistema propuesto por M. Álvarez et al. en [18] y [19].

Por otra parte, debido a que el uso de tecnologías de lado cliente puede utilizarse para "engañar" a los sistemas de crawling en su tarea, han aparecido varios trabajos relacionados con detectar lo que se conoce como Web Spam. Dentro del Web Spam existen diversas técnicas tales como el Cloacking [7] [20] [21] [22] o Redirection Spam [6] [22]. La primera de estas técnicas pretende detectar cuándo es un usuario normal y cuándo es un crawler el que realiza la petición de la página. Si el que realiza la petición es un motor de búsqueda el sitio Web mostrará un contenido diferente al que le mostraría si fuera un navegador de un usuario. Las técnicas de Redirection Spam pretenden ocultar las redirecciones para ser ejecutadas únicamente en un navegador. En ambos casos se consigue "mentir" al buscador de forma que indexe unos contenidos diferentes a los que realmente son.

Sin embargo, no se conocen escalas que permitan clasificar la efectividad de los sistemas de crawling respecto a su nivel de tratamiento de las tecnologías de la Web Oculta del lado cliente.

III. Tecnologías de lado cliente usadas en la construcción de sitios web

A continuación se enumeran las tecnologías de lado cliente más habituales en la creación de sitios web, normalmente usadas para mejorar la experiencia de usuario, generando contenido y enlaces de forma dinámica, en función de las acciones del usuario.

- JavaScript, dialecto del estándar ECMAScript, es un lenguaje imperativo y orientado a objetos. Permite la generación dinámica de la interfaz y su modificación en base a los eventos generados por el usuario, a través de una implementación del DOM [23].
- Applet [24], componente Java de una aplicación que se ejecuta en el cliente web. Permite tener acceso casi completo a la máquina, con velocidades similares a la de lenguajes compilados. Permite crear soluciones más escalables al número de usuarios.
- AJAX, conocida técnica de desarrollo que permite crear aplicaciones web interactivas. Utiliza JavaScript para el envío y la recepción de la petición/respuesta asíncrona del servidor, y normalmente JSON [25] como lenguaje para encapsular la información recibida.
- VbScript, lenguaje interpretado creado por Microsoft como variante del lenguaje Visual Basic. Se ha utilizado como parte esencial de aplicaciones ASP [26] y su funcionalidad es similar a la que aporta JavaScript.
- Flash, aplicación que permite crear interfaces vectoriales. La programación de dichas interfaces se hace mediante ActionScript, lenguaje de características parecidas a JavaScript y VbScript.

IV. Definición de la escala

A partir de las diferentes tecnologías descritas en el apartado anterior y del análisis de su uso por parte de los diseñadores de sitios web, se han identificado los siguientes tipos de ocurrencias:

- Enlaces de texto, que constituyen el nivel más básico de la escala.

```
<a href="a_11000100100000000000000000_test...html">Luis Ramirez Lucena</a>
```

- Navegaciones simples generadas con JavaScript, VbScript o ActionScript. Incluye enlaces generados mediante "document.write()" o funciones similares en otros lenguajes, que permiten añadir nuevos enlaces al HTML de forma dinámica.

```
<a href="JavaScript: ">Paolo Boi </a>
```

- Navegaciones generadas mediante un Applet, dentro de las cuales existen dos tipos a su vez, aquellas generadas a partir de un URL que se le pasa como argumento al Applet y aquellas otras cuyo URL está definido como una cadena en su código compilado.
- Navegaciones generadas mediante AJAX.
- Menús desplegables, generados mediante la ejecución de código script asociado a algún evento.
- Navegaciones generadas desde Flash. Existen dos tipos: aquellas que reciben el URL como argumento desde el código HTML y aquellas que lo tienen definido dentro del propio código ActionScript.
- Enlaces definidos como cadenas en ficheros .java, .class, u otro tipo de ficheros.
- Navegaciones generadas a partir de funciones que pueden estar definidas en algún lenguaje de script cuyo código puede estar embebido en el HTML o en un fichero externo.
- Navegaciones generadas mediante diferentes tipos de redirecciones:
  - Redirección en la etiqueta meta.
  - Redirección creada en el evento onLoad de la etiqueta body.
  - Redirección JavaScript, que se ejecutará en el momento en que se cargue la página.
  - Redirección creada sobre un Applet, al cargar la página con el Applet.
  - Redirección Flash, al procesar la página con su correspondiente fichero SWF.

De forma adicional, las navegaciones generadas con cualquiera de los métodos identificados pueden crear direcciones URL de tipo absoluto o relativo. Para las direcciones generadas a partir de cualquier lenguaje de script, se pueden distinguir los siguientes métodos de construcción:

- Una cadena estática dentro del Script.

```
menu_static_embedded_relative(){document.location="a_1001...html";}
```

- Una concatenación de cadenas.

```
function menu_concatenated_embedded_relative(){
var out="";out="a_1001001010000000000000000_test_menu"+
"_concatenated_embedded_relative.html"; document.location=out;}
```

- Ejecución de una función que construye en varios pasos el URL.

```
function menu_special_function_embedded_relative(){
var a1="win",a2="dow",a3=".location.",a4="replace",a5;
a5="('a_1001000110000000000000000_test_menu_special_function";
var a6="_embedded_relative.html')";var i,url="";
for(i=1;i<=6;i++) {url+=eval("a"+i);} eval(url);}
```

Por otra parte, los diferentes métodos enumerados pueden aparecer combinados. Por ejemplo, algunos sitios web construyen menús desplegables de forma dinámica, mediante la utilización de funciones "document.write()". El número de posibilidades es inabordable. Por este motivo, para este estudio se ha considerado un subconjunto reducido, pero suficientemente significativo, que se muestra en la Fig. 1. Consta de 70 niveles que representan los tipos básicos a partir de los cuales se podrían obtener el resto de casos, mediante combinaciones. El número de niveles considerados podría ser mayor, pero no aportaría más información ni sobre el uso en la Web de las tecnologías del lado cliente, ni tampoco sobre los métodos que usan los crawlers para descubrir enlaces. Siguiendo con el ejemplo de combinación comentado, no es necesario considerarlo en el estudio ya que se ha incluido de forma independiente el tratamiento de los menús y de navegaciones generadas con "document.write()". A partir de la información obtenida para los dos casos base, se podría concluir la capacidad del crawler para tratar el caso comentado.

Tipo de ruta	Tecnología	Ubicación	Tipo de cadena		Número
Relativa / Absoluta	Texto	Embebido	Estática		1 – 2
	JavaScript	Embebido / Externo	Estática / Concatenación / Función Especial		3 – 38
	Document.Write()				
	Menu JavaScript				
	Enlace en .java	Externo	Estática		39 – 40
	Enlace en .class	Externo	Estática		41 – 42
	Applet-Enlace HTML	Embebido	Estática		43 – 44
	Applet-Enlace Class	Externo	Estática		45 – 46
	Flash-Enlace HTML	Embebido	Estática		47 – 48
	Flash-Enlace SWF	Externo	Estática		49 – 50
	AJAX	Embebido	Estática		61 – 62
	JavaScript con marcador #	Embebido	Estática / Función Especial		63
					66
	VbScript	Embebido	Estática / Función Especial		67
					70
Relativa / Absoluta	Redirección	Externo	Estática	Etiqueta meta	51 – 52
				Etiqueta body	53 – 54
				JavaScript	55 – 56
				Applet	57 – 58
				Flash	59 – 60

Fig. 1. Combinaciones de los tipos de enlaces

A partir de los 70 niveles se propuso una primera agrupación en base a las tecnologías, los métodos de construcción de cadenas, la ubicación del código y los tipos de rutas del enlace. Se creó un sitio web (ver sección V) sobre el que se realizaron las pruebas mostradas en la sección VI. En base a los resultados obtenidos, se realizó un reagrupamiento de niveles teniendo en cuenta el número de crawlers que los trataron, consiguiendo de esta forma agrupar los niveles que presentaban igual dificultad para los crawlers, además de ordenarlos por complejidad.

La Fig. 2 muestra la escala propuesta. Está formada por 8 niveles, que representan de menor a mayor, la capacidad para tratar la Web Oculta del lado cliente. En los niveles más bajos se encuentran, entre otros, los enlaces de texto, redirecciones sencillas y enlaces de JavaScript generados con "document.write()" o menús con cadenas estáticas. En los niveles de dificultad alta aparecen, principalmente, enlaces en Applets, Flash, AJAX, VbScript o redirecciones complejas.

Nivel	Descripción	Escenarios probados
1	Enlace de texto	1,2
2	JavaScript/Document.Write/Menu – Cadena Estática – Embebida	3, 4, 15, 16, 27, 28
	JavaScript – Cadena concatenada – Embebida	6
3	Redireccion HTML/onBody/JavaScript	51, 52, 53, 54, 55, 56
	JavaScript con # – Cadena Estática – Embebida	63,64
	VbScript – Cadena Estática – Embebida	67,68
4	VbScript - Función Especial - Embebida	70
5	JavaScript/Document.Write – Cadena Estática – Externa/Embebida	9, 10, 18
	Document.Write/Menu – Cadena Estática – External	21, 22, 33, 34
	Menu – Cadena concatenada – Embebida	30
6	JavaScript/Document.Write/Menu – Cadena concatenada – External	12, 24, 36
	Applet – Cadena Estática en HTML	43,44
7	JavaScript – Cadena concatenada – Embebida – Relativa	5
	JavaScript – Función Especial – Embebida	7,8
	JavaScript – Cadena concatenada – Externa – Relativa	11
	JavaScript – Función Especial – Externa – Relativa	13,14
	Document.Write – Cadena concatenada – Embebida/Externa – Relativa	17,23
	Document.Write – Función Especial – Embebida/Externa	19, 20, 25, 26
	Menu – Cadena concatenada – Embebida/Externa – Relativa	29,35
	Menu – Función Especial – Embebida/Externa	31, 32, 37, 38
	Enlace en .class	41,42
	Enlace Ajax – Absoluta	62
8	Enlace .java	39,4
	Applet – Cadena Estática en .class	45,46
	Flash – Cadena Estática en HTML/SWF	47, 48, 49, 50
	Redirección Applet/Flash	57, 58, 59, 60
	Enlace Ajax – Relativa	61
	JavaScript con # – Función Especial – Embebida	65,66
	VbScript – Función Especial – Embebida	69

Fig. 2. Clasificación de los enlaces por dificultad

Una vez definida la escala, para poder clasificar los diferentes sistemas de crawling según el nivel de complejidad de los "enlaces" que traten, se proponen los siguientes métodos de evaluación:

- Media simple: trata todos los escenarios definidos, sin considerar su dificultad. Da una idea de qué crawlers tratan mayor número de escenarios y por tanto prestan mayor atención a la Web Oculta.
- Máximo nivel accedido: este modelo ordenará los crawlers según el nivel más alto de dificultad que son capaces de tratar. Se entiende que un crawler que obtiene un nivel máximo i tiene capacidad para procesar los enlaces de ese nivel e inferiores, aunque debido a motivos tales como bajo PageRank de la página web, pueda no llegar a analizarlos.
- Media ponderada: se asigna a cada escenario tratado un valor entre 0 y 1, que depende del número de crawlers que han sido capaces de procesarlo (0 cuando todos los crawlers han sido capaces de tratarlo). Este método indica qué crawlers serán los que puedan obtener mayor número de recursos de elevada dificultad en la Web Oculta de lado cliente, o recursos que la mayoría no alcanza.
- Ocho niveles: este modelo asigna 1 punto a cada nivel tratado. Si un crawler consigue procesar todos los escenarios de un nivel obtendrá 1 punto, es decir por cada escenario procesado con éxito consigue $1/n$, donde n es el número total de escenarios definidos para ese nivel.

Para los métodos de máximo nivel y ocho niveles, se obtienen valores entre 0 y 8, siendo 8 el nivel más alto. Para el resto de casos, se aplica una normalización para facilitar la comparación de resultados.

Aunque los métodos propuestos, por separado, no proporcionan una información concluyente respecto a la capacidad de los crawlers para el tratamiento de la Web Oculta del lado cliente, sí lo hacen si se consideran de forma conjunta, como puede verse en la sección VI de comparación de crawlers.

V. Sitio Web

Para poder comprobar cómo tratan los diferentes niveles los sistemas de crawling existentes, se ha creado un sitio web contra el que realizar los experimentos. En el sitio web "jstestingsite"[1], se han creado 70 enlaces, representando los 70 niveles definidos por la escala, usando y combinando las tecnologías explicadas. Con la finalidad de incentivar la indexación del sitio Web se han tomado diversas medidas, como incluirlo desde la página web del Departamento Tecnologías de la Información y las Comunicaciones, elaborar el contenido en inglés y añadir a cada nivel, la biografía de un ajedrecista. Esto último se ha hecho para evitar que los crawlers cataloguen la pagina como Web Spam. En la Fig. 3 se muestra la página principal del prototipo de sitio web. Se identifican las siguientes partes:

- En la parte superior, de izquierda a derecha, aparecen los enlaces en Menú JavaScript, los enlaces generados en el Applet y finalmente los enlaces en Flash.
- En el centro de la página aparece una tabla dividida en 4 columnas, primero el número que identificará a cada test, a continuación una pequeña descripción del test y finalmente el enlace relativo y absoluto.
- En la parte inferior, tras la tabla, aparece el contenido.

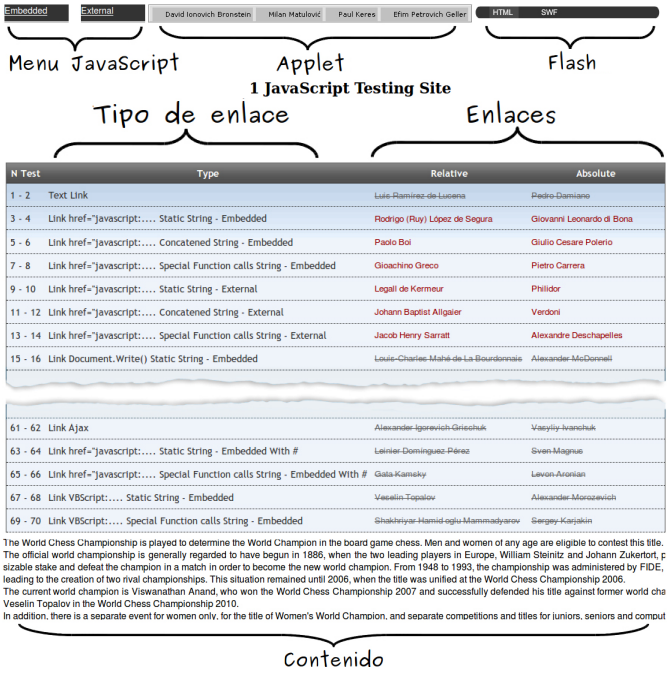

Fig. 3. Página principal del estudio

[1]http://www.tic.udc.es/~mad/resources/projects/jstestingsite/

Por otro lado se ha creado una página de resultado para cada una de las pruebas, de tal modo que si el crawler es capaz de acceder a dicho contenido significa que ha sido capaz de procesar el enlace. En la Fig. 4 se muestra una página de resultado como ejemplo, formada por los siguientes elementos:

- En la parte superior, en el centro, se muestra el número y nombre del test.
- En la parte superior izquierda aparece el código del test, una máscara binaria que representa numéricamente las características que tiene o no tiene la prueba.
- En el lateral izquierdo se muestra una tabla que enumera las características de la prueba.
- En la parte central se incluye la vida de un maestro ajedrecista.

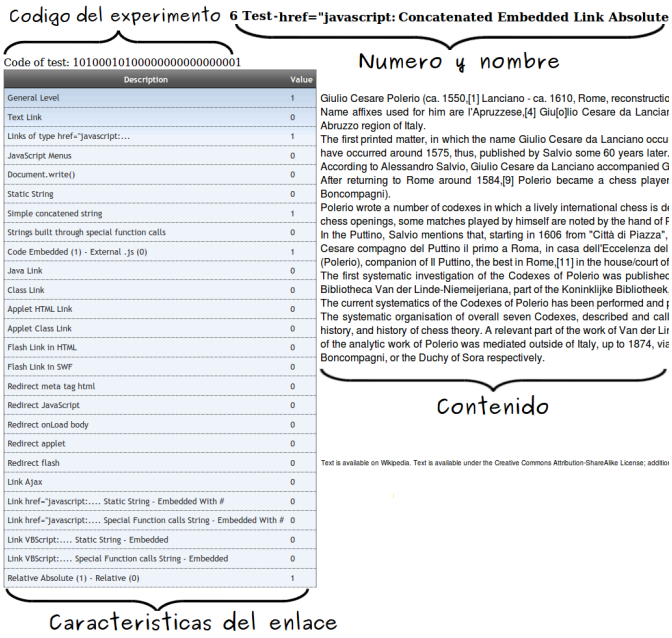

Fig. 4. Página de resultado de un test

VI. Resultados Experimentales

Esta sección incluye los experimentos realizados y los resultados obtenidos. Se han realizado pruebas contra el sitio web definido con crawlers OpenSource y comerciales, y con crawlers usados por los principales buscadores web. Para cada crawler se han obtenido dos tipos de resultados; información de indexación de contenido recopilado a través de los repositorios, en los que cada sistema almacena los documentos para permitir realizar búsquedas, e información de acceso recopilada a partir de los ficheros de log del servidor web del sitio crawleado. Tras esto se realiza una comparativa de los resultados de ambos tipos, atendiendo tanto al total de enlaces procesados como a los tipos de enlaces que han logrado tratar. Por último se presentan las puntuaciones obtenidas por cada crawler según los 4 métodos de evaluación definidos sobre la escala propuesta.

Para la ejecución de los crawlers OpenSource y comerciales se utilizaron las configuraciones que permitían maximizar el nivel de exploración de tecnologías del lado cliente. En el caso de los crawlers de buscadores web, en primer lugar se

analizaron los logs del servidor web que proporciona acceso al prototipo, para identificar los crawlers que estaban accediendo a alguna página del servidor. En base al listado publicado en la página oficial de robots [27], al User-Agent y dirección IP se determinó a qué buscadores pertenecen, y se eliminaron aquellos que no eran independientes, es decir que dependen del crawler de un tercero, y aquellos que forman parte de compañías de marketing. Para acelerar el proceso de visita e indexación se ha dado de alta manualmente el sitio web en los siguientes crawlers: Google[2], Bing[3], Yahoo![4], PicSearch[5] y Gigablast[6].
Para la generación automática de resultados de los logs del servidor, se implementó una herramienta que parte del listado de robots creado, añadiéndole las IPs y User-Agents de los crawlers OpenSource y comerciales, para automáticamente determinar qué crawlers han sido capaces de solicitar cada uno de los recursos expuestos por el servidor web.

A. Crawlers OpenSource y comerciales

Se realizó un análisis de 24 crawlers, que se muestran en la Fig. 5.

Crawler	Licencia
Advanced Site Crawler, Essential Scanner, GsiteCrawler, Heritrix, Htdig, ItSucks, Jcrawler, Jspider, Larbin, MnogoSearch, Nutch, Open Web Spider CS, Oss, Pavuk, Php Crawler, WebHTTrack	Free
JOC Web Spider, Mnogosearch, Visual Web Spider, Web Data Extractor, Web2Disk , WebCopier Pro	Shareware

Fig. 5. Crawlers OpenSource/Comerciales

Entre las características que comparten destacan: opciones de uso de proxy, autenticación (en ocasiones mediante formulario), limitaciones en número de documentos, diferentes protocolos, exclusión/inclusión de extensión de ficheros, opciones de dominio/directorio, cookies, User-Agent, Logging, trabajo con formularios, etc.

De esta lista se han seleccionado los 7 que mejores características presentan para el tratamiento de tecnologías de lado cliente. Entre los OpenSource, se seleccionaron los siguientes:

- Nutch [8], crawler y buscador basado en Lucene. Desarrollado en Java y con arquitectura basada en Hadoop, lo que permite su implantación como crawler distribuido. Permite extraer enlaces tanto de código JavaScript como de Flash, aunque utilizando heurísticas muy sencillas.
- Heritrix [9], un crawler conocido por ser usado por Internet Archive[7]. En lo que refiere al tratamiento de JavaScript lo hace mediante el uso de expresiones regulares, de forma muy similar a como lo hace Nutch.
- Pavuk [28], crawler que destaca por permitir rellenar formularios de forma automática e intentar obtener enlaces en JavaScript mediante el uso de expresiones regulares.
- WebHTTrack, crawler escrito en C que permite analizar ficheros Java, Flash e intenta obtener enlaces en JavaScript a partir de diferentes heurísticas. A diferencia de los anteriores realiza comprobaciones en busca de cadenas tales como “Object.Write”, “document.location”, “window.replace”, etc.

De los crawlers comerciales se seleccionaron:

- Teleport [29], crawler con cierto reconocimiento sobre todo en la década de los 90, y que incluye soporte para análisis de JavaScript.
- Web2Disk [30], crawler desarrollado por la empresa Inspyder. Permite un análisis básico o avanzado de JavaScript con su correspondiente diferencia en resultados y en tiempo de cómputo.
- WebCopierPro [31], crawler que permite procesar eventos dinámicos, analizar código JavaScript y Flash en búsqueda de enlaces.

B. Resultados de crawlers OpenSource y comerciales

En primer lugar se analizan los resultados en función del contenido del sitio web que ha sido indexado por los diferentes crawlers OpenSource y comerciales. Esto tiene lugar analizando los diferentes repositorios generados por los crawler durante su operación. El crawler que mejores resultados obtiene es WebCopierPro (tabla de la izquierda de la Fig. 6) con un 57,14% de los niveles procesados, seguido de Heritrix con 47,14% y Web2Disk con 34,29%. Pocos de ellos obtienen valores por encima de 25% en la mayoría de los tipos de enlaces, no siendo capaces de obtener enlaces en muchos casos. Del mismo modo es importante observar los bajos resultados obtenidos en el apartado de redirecciones, sobre todo en el caso de WebCopierPro que no es capaz de tratar ninguna, cuando en niveles de mayor complejidad obtiene resultados del 100%. Ninguno de los crawlers alcanza el 100% en las redirecciones. Esto es debido a que ninguno de ellos ha sido capaz de procesar páginas con redirecciones incrustadas en Applets o Flash. Son capaces de descargar la página, pero no ejecutan el Flash o Applet que genera la redirección.

Analizando los resultados según el tipo de enlace, mostrados en la Fig. 7, se obtiene una visión sobre qué tipos de enlaces son procesados por un mayor número de crawlers. Sin prestar atención al 100% que consiguen para los enlaces de texto se observa que: Consiguen atravesar entre el 35% y el 40% de los niveles de Href=”javascript ; “document.write()” ; Menu links ; Links with # y VbScript. Muy por debajo está el 7% conseguido en los enlaces de ficheros class o Java y aquellos generados mediante AJAX. Ninguno ha conseguido en enlaces de Flash, puede ser debido a la poca atención de los crawlers sobre estos tipos de enlaces o a la dificultad que entraña obtenerlos.

Como se muestra en la Fig. 8, debido a que la mayoría de los crawlers trabajan buscando URL’s con expresiones regulares, el porcentaje de enlaces encontrados cae de un 42.52% en los considerados enlaces estáticos a un 27.38% en el caso de enlaces generados mediante la concatenación de cadenas y finalmente a un 15.18% en el caso de enlaces generados con funciones. Se concluye que la probabilidad de encontrar enlaces mediante expresiones regulares o tratamiento de texto es inversamente proporcional a la dificultad con la que se genera el enlace.

Resumiendo los datos de la Fig. 6 se puede decir que sólo una tercera parte de los enlaces generados con tecnologías

[2]http://www.google.es/addurl/
[3]http://www.bing.com/webmaster/SubmitSitePage.aspx
[4]http://siteexplorer.search.yahoo.com/submit
[5]http://www.picsearch.com/menu.cgi?item=FAQ
[6]http://www.gigablast.com/addurl
[7]http://www.archive.org/

Tipo de enlace	REPOSITORIOS													LOGS			
	Heritrix		Nutch		Pavuk		Teleport		Web2Disk		WebCopierPro		WebHTTrack	Teleport		WebHTTrack	
Texto	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**	2	**100,00%**
Href="javaScript:..."	6	**50,00%**	3	**25,00%**	0	**0,00%**	3	**25,00%**	5	**41,67%**	12	**100,00%**	3	**25,00%**	4	**33,33%**	3	**25,00%**
Document.write	6	**50,00%**	3	**25,00%**	0	**0,00%**	2	**16,67%**	4	**33,33%**	12	**100,00%**	2	**16,67%**	3	**25,00%**	2	**16,67%**
Menu	6	**50,00%**	3	**25,00%**	0	**0,00%**	2	**16,67%**	4	**33,33%**	12	**100,00%**	2	**16,67%**	3	**25,00%**	2	**16,67%**
Flash	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	2	**50,00%**
Applet	2	**50,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	2	**50,00%**	0	**0,00%**	2	**50,00%**
Redirecciones	6	**60,00%**	6	**60,00%**	2	**20,00%**	6	**60,00%**	4	**40,00%**	0	**0,00%**	6	**60,00%**	6	**60,00%**	6	**60,00%**
Class/Java	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	2	**50,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**
Ajax	0	**0,00%**	1	**50,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**	0	**0,00%**
Mediante #	2	**50,00%**	2	**50,00%**	0	**0,00%**	2	**50,00%**	2	**50,00%**	0	**0,00%**	2	**50,00%**	3	**75,00%**	2	**50,00%**
VbScript	3	**75,00%**	3	**75,00%**	0	**0,00%**	3	**75,00%**	3	**75,00%**	0	**0,00%**	2	**50,00%**	3	**75,00%**	2	**50,00%**
Enlaces cadena estática	26	**61,90%**	19	**45,24%**	4	**9,52%**	18	**42,86%**	22	**52,38%**	16	**38,10%**	20	**47,62%**	18	**42,86%**	22	**52,38%**
Enlaces cadena concatenada		**50,00%**		**16,67%**		**0,00%**		**8,33%**		**8,33%**		**100,00%**		**8,33%**	1	**8,33%**	1	**8,33%**
Enlaces de funciones espiales		**6,25%**		**12,50%**		**0,00%**		**6,25%**		**6,25%**		**75,00%**		**0,00%**	5	**31,25%**	0	**0,00%**
Enlaces totales conseguidos	33	**47,14%**	23	**32,86%**	4	**5,71%**	20	**28,57%**	24	**34,29%**	40	**57,14%**	21	**30,00%**	24	**34,29%**	23	**32,86%**

Fig. 6. Resumen de resultados de crawlers OpenSource y comerciales. A la derecha en los repositorios y a la izquierda en los logs

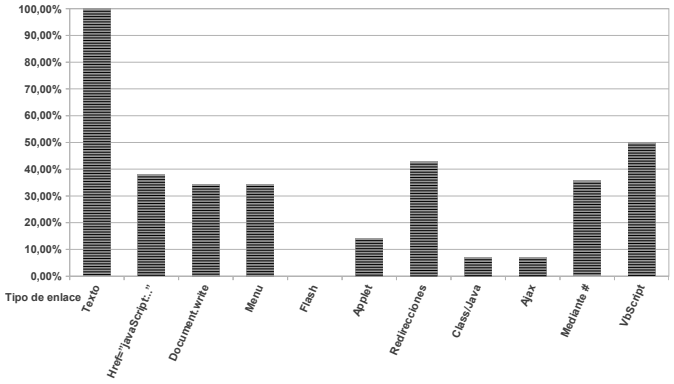

Fig. 7. Tipos de enlaces extraídos por los Crawlers OpenSource y comerciales

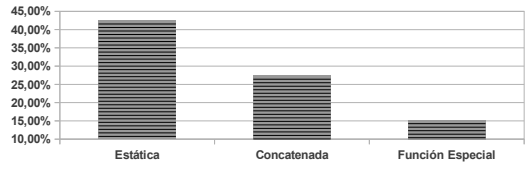

Fig. 8. Crawlers OpenSource y comerciales según el tipo de cadena

del lado cliente son tratados por los crawlers OpenSource o comerciales.

Como complemento a los resultados presentados analizando los índices, se han analizado las entradas que generaba cada crawler en los logs del servidor donde estaba almacenado el sitio Web. Con esto se pretende analizar más en detalle cómo ”intentan“, los crawlers, procesar cada nivel. La tabla de la derecha de la Fig. 6 muestra las diferencias encontradas con los resultados obtenidos al analizar los repositorios de cada crawler, mostrados en la tabla de la izquierda de la misma figura.

- WebHTTrack, intentó acceder a los niveles de Flash en que las direcciones están en el código HTML para ser pasadas a Flash mediante parámetros. No tuvo en cuenta que la estructura para pasar parámetros a Flash es ”parametro1=datos1&parametro2=datos2&....” por lo que considera toda la cadena como un único URL cuando no lo es.

```
"GET /~mad/resources/projects/jstestingsite/1_test/link_relative=a_1000010010
0001000000000000_test_flash_link_html_relative.html&link_absolute=http://www.
tic.udc.es/~mad/resources/projects/jstestingsite/1_test/a_10000100100001000000
000001_test_flash_link_html_absolute.html"
```

- Teleport intentó acceder a URLs absolutos generados mediante una función. Detecto “http...” y busco el correspondiente “.html” que indica el final de la dirección. No obstante, no detecta que por el medio se han definido variables, y por tanto el URL generado es incorrecto.

```
"GET /~mad/resources/projects/jstestingsite/1_test/";var%20a6="a_1000100110000
0000000000001_test_document_write_function_special_embedded_absolute.html"
```

C. Resultados de crawlers de los principales buscadores web

Para realizar el análisis de crawlers utilizados por los principales buscadores web, se han esperado 70 días desde que se puso accesible el sitio web de prueba y se dio de alta en los diferentes buscadores. Tras este tiempo, se comprobó, en los logs del sitio web, qué crawlers de los contemplados habían accedido al menos a la página principal del sitio web. La Fig. 9 contiene los resultados obtenidos. No se muestran los resultados de Alexa, Ask, Bing, Gigablast y PicSearch, debido a que no indexaron el sitio web. Solo Google y Yahoo! indexaron el sitio web. Entre los factores que pudieron motivar que ciertos crawlers no indexaran el sitio web se puede considerar:

- Bajo PageRank.
- Alta profundidad en donde se encuentra alojado.
- Alto contenido de código, lo que implica que en base a una heurística un crawler determine la posibilidad de que el sitio web contenga Malware o bien determine que los enlaces tienen una alta dificultad para ser procesados.
- Contenido insuficiente o de mala calidad.

Google consiguió procesar un 44.29% de los enlaces, lo cual indica que no ha sido por el diseño, contenido o demás variables por las que el sitio no ha sido indexado por los demás. El crawler de Yahoo! también ha procesado algún enlace, pero de escasa dificultad.

Aunque GoogleBot consiguió procesar el 50% de muchos de los niveles propuestos, se observó que no logró un 100% debido a que en la otra mitad de los enlaces el código estaba almacenado en un fichero externo que no llegó a procesar. En caso contrario GoogleBot conseguiría resultados mucho mejores. Las tecnologías usadas en los enlaces que no procesó eran Flash, Applets y AJAX, del mismo modo que tampoco entró a analizar ficheros .class o .java.

En el caso de estos crawlers no se muestran por separado los resultados en los índices, es decir en sus sistemas de búsqueda, y en los logs del servidor del sitio web, ya que de ambas formas se obtienen resultados idénticos.

Tipo de enlace	Google		Yahoo	
Texto	2	100,00%	1	50,00%
Href="javaScript:..."	6	50,00%	0	0,00%
Document.write	6	50,00%	0	0,00%
Menu	6	50,00%	0	0,00%
Flash	0	0,00%	0	0,00%
Applet	0	0,00%	0	0,00%
Redirecciones	6	60,00%	2	20,00%
Class/Java	0	0,00%	0	0,00%
Ajax	0	0,00%	0	0,00%
Mediante #	4	100,00%	0	0,00%
VbScript	1	25,00%	0	0,00%
Enlaces cadena estática	17	40,48%	3	7,14%
Enlaces cadena concatenada	6	50,00%	0	0,00%
Enlaces de funciones espiales	8	50,00%	0	0,00%
Enlaces totales conseguidos	31	44,29%	3	4,29%

Fig. 9. Resumen resultados crawler de buscadores

D. Comparación de resultados

En la Fig. 10 se muestra un resumen comparativo de resultados. Claramente los crawlers OpenSource y comerciales obtienen en general mejores resultados. Es Google únicamente quien consigue unos números similares. Esto puede ser debido a que un crawler OpenSource o comercial está configurado por un usuario que indica el tipo de enlaces que debe seguir sin preocuparse de aspectos de rendimiento o seguridad, circunstancias que un crawler global de un buscador debe tener muy en cuenta.

En la Fig. 11 se comparan los resultados en base a la tecnología usada en la construcción del enlace. Nuevamente se observa un mejor comportamiento de los crawlers OpenSource. La curva que realiza cada grupo de crawlers es similar por lo que se puede concluir que a pesar de conseguir un menor número de enlaces de cada tecnología, sí muestran el mismo interés por procesar el mismo tipo de tecnologías.

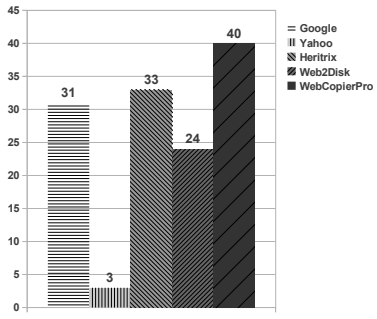

Fig. 10. Comparativa crawlers de motores de búsqueda y OpenSource

E. Clasificación de los crawlers en base a la escala definida

La Fig. 12 muestra el resultado de clasificar los distintos sistemas de crawling en base a la escala definida, según los diferentes métodos de evaluación propuestos.

- En media simple, WebCopier obtiene los mejores resultados, por delante de Heritrix y Google.
- Para el modelo de máximo nivel, Google se desmarca del resto consiguiendo procesar enlaces de nivel 8, seguido por WebCopier que obtiene un 7 y Heritrix un 6. Que Google haya alcanzado el nivel máximo según este modelo y no en otros, indica que podría tener capacidad para tratar cualquiera de los escenarios considerados, pero no lo hace debido a políticas internas.
- Para el método de la media ponderada, nuevamente WebCopier, seguido de Google, Heritrix y Nutch presentan los mejores resultados.
- En ocho niveles, Google cede los primeros puestos a Heritrix, Web2Disk y WebCopier. Esto indica que estos tres o bien han tratado gran cantidad de niveles de cada grupo o bien han atravesado enlaces que formaban parte de un grupo con pocos enlaces, lo cual da una alta puntuación a cada enlace procesado.

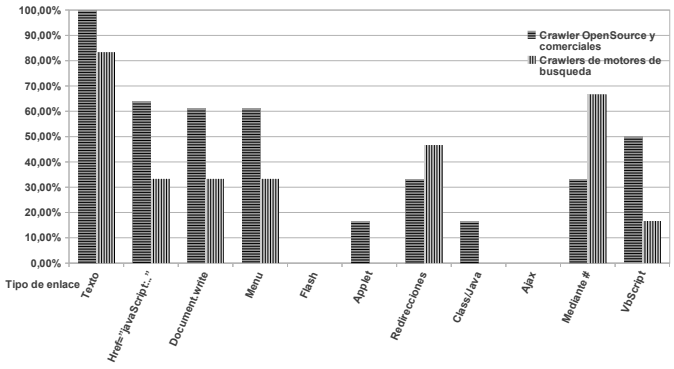

Fig. 11. Comparativa resultados de crawlers según el tipo de link

Se puede concluir que los crawlers que más y mejor tratan la Web Oculta del lado cliente son Google y WebCopier, seguidos por Heritrix, Nutch o Web2Disk. Es importante resaltar los resultados obtenidos para GoogleBot, por tratarse de un sistema de crawling orientado a toda la Web, con grandes requisitos de rendimiento y seguridad, y que no por ello ha descuidado el tratamiento de este tipo de tecnologías.

	Heritrix	Nutch	Pavuk	Teleport	Web2Disk	WebCopier	WebHTTRack	Google	Yahoo
Media simple	3,77	2,63	0,46	2,29	2,74	4,57	2,40	3,54	0,34
Máximo nivel	6,00	7,00	3,00	4,00	5,00	7,00	6,00	8,00	3,00
Media ponderada	1,63	0,99	0,14	0,75	1,03	3,42	0,86	2,34	0,11
Ocho niveles	6,00	4,30	1,20	4,00	4,55	4,55	3,40	3,70	0,70

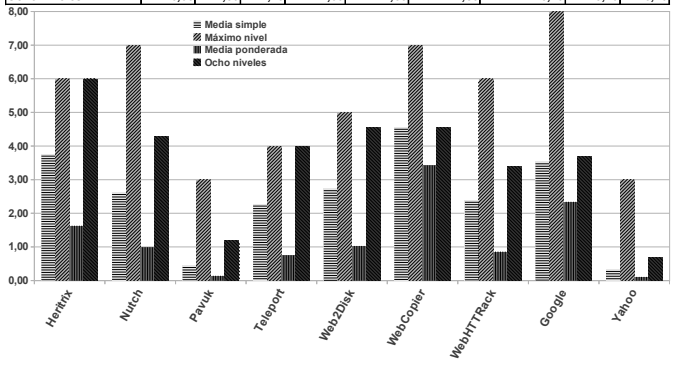

Fig. 12. Resultados en las escalas propuestas

VII. Conclusiones

Este artículo propone una escala que tiene en cuenta los diferentes niveles de dificultad existentes en los sitios de la Web Oculta del lado cliente. La definición de la escala permite clasificar los sistemas de crawling en base a su efectividad accediendo a dicha Web Oculta. La escala se ha definido en base a las diferentes tecnologías del lado cliente que

actualmente se usan por parte de los diseñadores para la creación de sitios web y a la dificultad que cada una de ellas presentan a los crawlers. Existen trabajos previos como el de M. Weideman y F. Schwenke en [12], pero en donde sólo se analizaban enlaces con JavaScript. En este trabajo se contempla un abanico de tecnologías mayor y más actuales. Por otro lado también se han incluido en el estudio crawlers OpenSource y comerciales con soporte para el tratamiento de las tecnologías del lado cliente.

Para realizar la clasificación de los diferentes sistemas de crawling, se ha creado un sitio web implementando las diferentes dificultades incluidas en la escala.

Los resultados obtenidos muestran que tanto en los niveles logrados como en los intentados, la mayor parte de las veces los crawlers tratan de descubrir los URL's procesando el código como texto, utilizando expresiones regulares. Es cierto que esto permite descubrir gran cantidad de escenarios y que el coste computacional es menor, pero se ha visto que la mayoría de las direcciones que forman parte de las tecnologías del lado cliente no son descubiertas. Los únicos para los que se puede concluir lo contrario son WebCopier y Google, que seguramente harán uso de algún intérprete que les permita ejecutar código.

Para tratar las dificultades presentadas por los niveles actualmente ignorados por los sistemas de crawling convencionales (tanto OpenSource como comerciales) existen diferentes aproximaciones en la literatura, que según los experimentos realizados no se utilizan en el práctica, seguramente por razones de eficiencia. Es el caso de la utilización de mini-navegadores web como componentes de crawling, como aparece descrito en [18], que permiten simular la navegación que realizaría un usuario al entrar en un sitio web.

Para finalizar, es importante resaltar que el tratamiento de la Web Oculta del lado cliente presenta desafíos como Redirection Spam o Cloacking. Una correcta detección evitaría perjuicios al usuario final y motivaría a que los crawlers presten atención a dichas páginas sin correr el riesgo de que éstas contengan algún tipo de Malware o redirección no deseada.

VIII. Trabajos futuros

Entre los estudios que se proponen como continuación de este trabajo está la mejora de los métodos de evaluación en base a la escala, para que tengan en cuenta el porcentaje de uso en la Web actual de los diferentes niveles de dificultad. Entre otras cosas permitirá determinar el volumen de información que se está quedando fuera del alcance de los crawlers, por no tratar algunos de los escenarios, y con ello determinar la importancia real del análisis de la Web Oculta del lado cliente.

También se plantea estudiar cómo afectan las diferentes características de un sitio web a su indexación y al análisis que los crawlers hacen de su contenido para localizar nuevos enlaces. De esta forma se podría determinar si la importancia, la temática y el número de visitas de un sitio afecta a cómo son tratados por los crawlers.

Por último, y debido a que los crawlers que han obtenido mejores resultados no tienen su código público, se podría considerar el diseño de algoritmos que permitan extraer enlaces de las tecnologías mostradas sin necesidad de utilizar mini-navegadores ni intérpretes completos, para que sean capaces de escalar a un crawling global.

Agradecimientos

Este trabajo de investigación ha sido financiado por el Ministerio de Educación y Ciencia de España y los fondos FEDER de la Unión Europea (Proyecto TIN2009-14203).

Referencias

[1] M. K. Bergman, "The deep web: Surfacing hidden value," 2000.
[2] Mozilla, "Javascript - mcd centro de documentacion."
[3] A. T. H. III, *Ajax: The Definitive Guide*. O'Reilly Media, 2008.
[4] Microsoft, "Vbscript user's guide," 2011. [Online; accessed 18-February-2011].
[5] Adobe, "application programming — adobe flash platform," 2011. [Internet; accessed 11-marzo-2011].
[6] K. Chellapilla and A. Maykov, "A taxonomy of javascript redirection spam," in *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, (New York, NY, USA), pp. 81–88, ACM, 2007.
[7] B. Wu and B. D. Davison, "Cloaking and redirection: A preliminary study," 2005.
[8] R. Khare and D. Cutting, "Nutch: A flexible and scalable open-source web search engine," tech. rep., 2004.
[9] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic, "Introduction to heritrix, an archival quality web crawler," in *4th International Web Archiving Workshop (IWAW04)*, 2004.
[10] "W3Techs - World Wide Web Technology Surveys." http://w3techs.com/, 2011. [Online; accessed 22-March-2011].
[11] "BuiltWith Web Technology Usage Statistics." http://trends.builtwith.com/, 2011. [Online; accessed 22-March-2011].
[12] M. Weideman and F. Schwenke, "The influence that JavaScript has on the visibility of a Website to search engines - a pilot study," *Information Research*, vol. 11, Jul 2006.
[13] S. Raghavan and H. Garcia-Molina, "Crawling the hidden web," in *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, (San Francisco, CA, USA), pp. 129–138, Morgan Kaufmann Publishers Inc., 2001.
[14] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl," *Proc. VLDB Endow.*, vol. 1, pp. 1241–1252, August 2008.
[15] F. C. A. P. Manuel Ãlvarez, Juan Raposo, "Crawling web pages with support for client-side dynamism," (University of A Coruna, 15071 A Coruna, Spain), 2006.
[16] Mozilla, "Mozilla rhino javascript engine," 2011. [Online; accessed 18-February-2011].
[17] "v8 - V8 JavaScript Engine." http://code.google.com/p/v8/, 2011. [Online; accessed 21-March-2011].
[18] M. Á. Díaz, *Arquitectura para Crawling Dirigido de Información Contenida en la Web Oculta*. PhD thesis.
[19] F. C. A. P. Manuel Ãlvarez, Juan Raposo, "A task-specific approach for crawling the deep web," *Journal Engineering Letters. Special Issue:*.
[20] B. Wu and B. D. Davison, "Detecting semantic cloaking on the web," in *Proceedings of the 15th International World Wide Web Conference*, pp. 819–828, ACM Press, 2006.
[21] B. Wu and B. D. Davison, "Identifying link farm spam pages," in *Proceedings of the 14th International World Wide Web Conference*, pp. 820–829, ACM Press, 2005.
[22] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," 2005.
[23] "The w3 consortium the document object model." http://www.w3.org/DOM/, 2011. [Online; accessed 18-February-2011].
[24] Wikipedia, "Applet — wikipedia, the free encyclopedia," 2011. [Online; accessed 18-February-2011].
[25] "JSON." http://json.org/, 2011. [Online; accessed 22-March-2011].
[26] Microsoft, "The official microsoft asp.net site," 2011. [Online; accessed 18-February-2011].
[27] "The Web Robots Pages." http://www.robotstxt.org/, 2011. [Online; accessed 18-February-2011].
[28] "Pavuk Web page." http://www.pavuk.org/, 2011. [Online; accessed 18-February-2011].
[29] "Teleport Web page." http://www.tenmax.com/teleport/pro/home.htm, 2011. [Online; accessed 18-February-2011].
[30] "Web2Disk Web page." http://www.inspyder.com/products/Web2Disk/Default.aspx, 2011. [Online; accessed 18-February-2011].
[31] "Web Copier Pro Web page." http://www.maximumsoft.com/products/wc_pro/overview.html, 2011. [Online; accessed 18-February-2011].