FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

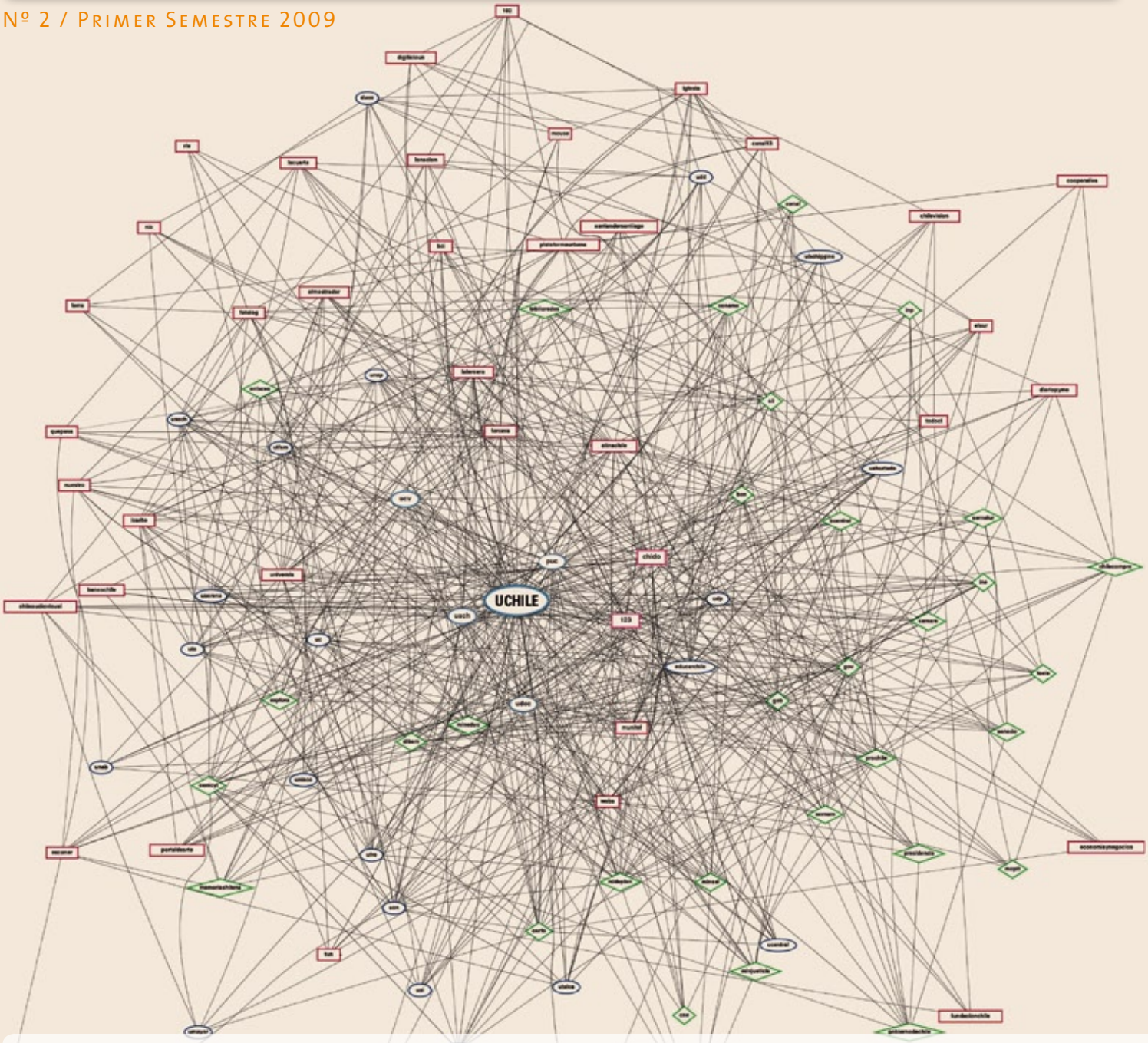
REVISTA

BITS de Ciencia

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

UNIVERSIDAD DE CHILE

Nº 2 / PRIMER SEMESTRE 2009



REVISTA
BITS de Ciencia
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Dirección

Departamento de Ciencias
de la Computación
Avda. Blanco Encalada 2120, 3º piso
Santiago, Chile
837-0459 Santiago
www.dcc.uchile.cl
Teléfono: 56-2-9780652
Fax: 56-2-6895531
dcc@dcc.uchile.cl

CONTENIDOS

EDITORIAL

La buena recepción por parte de la comunidad al primer número de la Revista Bits de Ciencia nos ha impulsado a ser bastante más ambiciosos en este segundo número. Esto ha implicado tres cambios fundamentales. El primero salta a la vista, y es que nos hemos preocupado mucho más del formato. Queremos así responder positivamente a algunos comentarios de nuestros alumnos expresando que el primer número parecía más un artículo científico (escrito en Latex, por cierto) que una revista propiamente tal. Nos pareció que tenían razón, y es por eso que nos hemos vestido de pantalón largo. En particular, la diagramación ha sido hecha profesionalmente, mejoramos la calidad del papel e introdujimos el color.

El segundo cambio es que de ahora en adelante cada número de la revista estará dedicado a un tema en particular. Para este segundo número elegimos el tema de la Ciencia de la Web en Chile. Para ello invitamos a varios de los expertos nacionales del área a participar. Incluimos respecto de este tema el estudio de Ricardo Baeza sobre la web en Chile, el de Claudio Gutiérrez y Mauricio Marín sobre el desarrollo de la ciencia de la web en el país, el survey de Mauricio Monsalve sobre redes sociales, además de un artículo de José Miguel Piquer sobre la primera conexión a internet en Chile. Sin embargo, y para no darle un giro completamente monotemático a la revista, sumamos también en este número otros artículos de interés general.

El tercer cambio tiene que ver con la estructuración del contenido de cada número de la revista.

Introdujimos secciones que dividan los diferentes tipos de artículos, y que sirvan como referencia estructural para los siguientes números. Por supuesto, este patrón no es totalmente rígido, y es probable que en el futuro agregemos nuevas secciones mientras algunas de las actuales desaparezcan.

Finalmente, me gustaría responder a nombre del Comité Editorial una pregunta que naturalmente surgió dentro del Departamento luego del primer número: ¿A quién está dirigida esta revista? Nos encantaría responder que a la comunidad informática en general. Pero sabemos que eso es imposible porque no tenemos ni la experiencia ni el tiempo para hacerlo. Nuestra respuesta es bastante más modesta: A los académicos interesados en la computación en Chile, a nuestros alumnos y a todos los alumnos de ciencia de la computación del país (y, por qué no, también del cono sur). Por supuesto, esto sin perjuicio de querer dirigirnos también a cualquier otra persona –empresario, funcionario de gobierno, científico, etc.– que sienta afinidad por el área y tenga amor por el mundo académico.

La revista está abierta a todos ellos también.

Profesor Pablo Barceló
Editor Revista Bits de Ciencia

Regreso al Futuro:
Depuración Omnisciente

La depuración es una actividad tediosa y costosa que requiere una comprensión profunda del comportamiento dinámico de los programas. Un depurador omnisciente, debido a que permite navegar en forma transparente en el historial de ejecución de los programas, facilita la localización de la causa raíz de los errores. ¿Por qué, entonces, no todos tenemos un depurador omnisciente en nuestro entorno de desarrollo favorito? Por cierto, para hacer práctica la depuración omnisciente es necesario superar varios desafíos, pero ¿son acaso estos una barrera definitiva para su adopción? Este artículo describe TOD, nuestro depurador omnisciente escalable para Java. TOD se integra en el ambiente de desarrollo Eclipse y contribuye a hacer práctica la depuración omnisciente.

Guillaume Pothier
Estudiante de Doctorado en Ciencias mención Computación, DCC, Universidad de Chile, bajo la supervisión del profesor Éric Tanter. Ingeniero en Ciencia de la Computación, École des Mines de Nantes, Francia.
gpothier@dcc.uchile.cl

Éric Tanter
Profesor Asistente, DCC, Universidad de Chile. Ph.D. en Computer Science, Universidad de Nantes y Universidad de Chile. Lidera el Laboratorio PLEIAD (Programming Languages and Environments for Intelligent, Adaptable, and Distributed Systems).
etanter@dcc.uchile.cl

INTRODUCCIÓN

La depuración representa una parte importante del costo del desarrollo de software. Un estudio del NIST (National Institute of Standards and Technology) de 2002 muestra que los errores de software tienen un costo enorme sobre la economía de EE.UU. [1] y menciona que *"los desarrolladores ya gastan aproximadamente 80 por ciento de los costos en identificar y corregir defectos"*. En un estudio empírico de "hazañas" de depuración,

Marc Eisenstadt determinó que la principal causa de la dificultad de encontrar los errores es la distancia temporal y espacial entre la *causa raíz* y el *síntoma* del error [2]; una vez que un error está precisamente localizado, arreglarlo es a menudo trivial. Desafortunadamente, la mayoría de los depuradores en uso hoy en día otorgan una ayuda muy limitada con respecto a la navegación temporal; los programadores deben con frecuencia simular mentalmente la ejecución de sus programas.

Los depuradores omniscientes mejoran de sobremanera esa situación, permitiendo a los programadores navegar fácilmente hacia adelante y atrás en el historial de ejecución de un programa, así como encontrar de inmediato la causa raíz de los errores, gracias a *vínculos causales* que pueden ser atravesados hacia atrás en el tiempo [3]. Por lo tanto, un depurador omnisciente puede tener un gran impacto sobre la eficiencia del proceso de desarrollo.

La depuración omnisciente no es, desde luego, una idea nueva: el primer depurador omnisciente, EXDAMS [4], fue creado en 1969. Sin embargo, a pesar de las numerosas propuestas que se han hecho desde entonces, los depuradores omniscientes todavía no forman parte del típico ambiente de desarrollo. ¿Serán acaso los desafíos de la depuración omnisciente una barrera definitiva para su adopción?

LA DEPURACIÓN OMNISCIENTE EN POCAS PALABRAS

Enfoques tradicionales de depuración

Existen dos enfoques tradicionales para la depuración: basada en registros, o *logs*, y basada en puntos de quiebre, o *breakpoints* (Figura 1). El primer enfoque consiste en insertar instrucciones de registro en el código fuente para producir una bitácora ad-hoc durante la ejecución del programa. Esa técnica revela efectivamente el historial de ejecución del programa, pero tiene serios inconvenientes: requiere modificaciones engorrosas, extendidas y anticipadas del código fuente, y no es escalable cuando las bitácoras deben ser analizadas manualmente.

El segundo enfoque consiste en correr el programa con una herramienta de depuración dedicada, que da al programador la posibilidad de detener la ejecución en determinados *breakpoints*, inspeccionar el contenido de la memoria, y seguir la ejecución paso a paso. Desafortunadamente, cuando la ejecución está detenida, la información acerca de estados y actividades anteriores del programa está limitada a la que está accesible desde la pila de activaciones. Por lo tanto, los desarrolladores que usan depuradores basados en breakpoints deben con frecuencia volver a ejecutar el programa entero con distintos breakpoints para progresivamente acercarse a la causa del error.

La depuración omnisciente

Un depurador omnisciente registra en forma automática el historial completo, o huella de ejecución, del programa depurado, y permite al usuario explorarlo libremente (Figura 1). Este enfoque combina las ventajas de la depuración basada en registros (la información sobre la actividad pasada no se pierde) y las de la depuración basada en breakpoints (navegación interactiva, ejecución paso a paso, inspección de la pila de activación completa). Los depuradores omniscientes simulan la ejecución paso a paso hacia adelante y *atrás*, haciendo

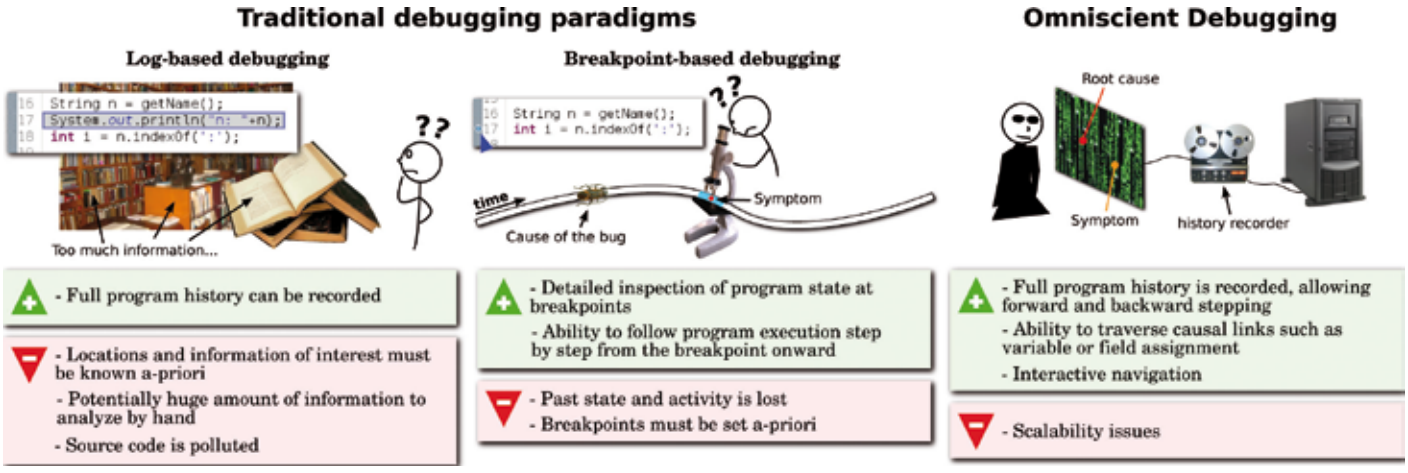


Fig. 1 Enfoques de depuración.

innecesario volver a ejecutar el programa varias veces para encontrar la causa raíz de un error. Y sobre todo, permiten atravesar vínculos causales, respondiendo instantáneamente a preguntas como *"¿Cuándo adquirió la variable x el valor null?"*, o *"¿cuál era el estado del objeto o cuándo fue pasado como argumento al método foo?"*

Desafíos

A pesar de sus claras ventajas sobre los enfoques tradicionales, la depuración omnisciente todavía no es considerada realista a causa de importantes problemas de escalabilidad:

1. La captura de una huella de ejecución causa una sobrecarga sobre el programa depurado, reduciendo su eficiencia.
2. Las huellas de ejecución llegan rápidamente a ser muy pesadas, por lo tanto requieren un sistema de almacenamiento rápido y escalable.
3. Las consultas sobre una huella de ejecución posiblemente muy pesada deben ser procesadas con suficiente prisa para poder garantizar la rapidez de las interacciones con el usuario.
4. Cualquier sea el peso de la huella de ejecución, el desarrollador debe siempre poder localizar rápidamente puntos de interés, y establecer relaciones significativas entre distintos puntos.

ARQUITECTURA DE TOD

TOD [5] es un depurador omnisciente para Java integrado en el ambiente de desarrollo Eclipse. La figura 2 delinea sus principios de operación:

1. **Instrumentación:** cuando una clase está a punto de ser cargada por la JVM, el agente envía su bytecode al tejedor (*weaver*), que inserta código de generación de eventos en la clase, y luego devuelve la versión modificada a la JVM.
2. **Generación de eventos:** A medida que el programa instrumentado se ejecuta, se generan eventos y se envían a la base de datos. La secuencia de eventos generados constituye la huella de ejecución.
3. **Almacenamiento e indexación:** La base de datos altamente especializada almacena los eventos a un ritmo muy alto, y al mismo tiempo los indexa para permitir un procesamiento rápido de las consultas. Además, una base de datos estructural almacena información sobre la estructura estática del programa depurado, tal como sus clases y métodos.
4. **Consultas y navegación:** El desarrollador navega en la huella de ejecución a través de la interfaz del depurador, que está integrada en el ambiente Eclipse.

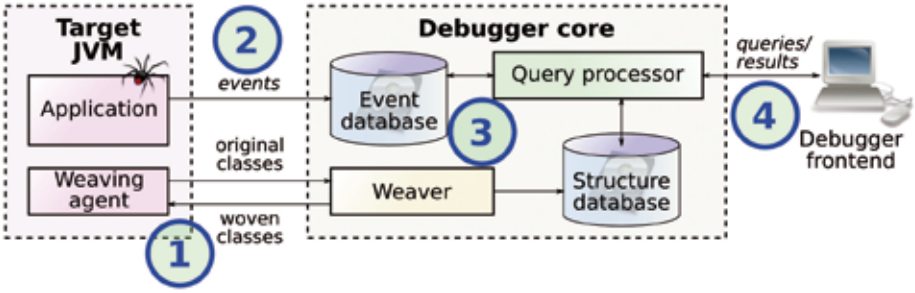


Fig. 2 Cómo funciona TOD: arquitectura y operación.

Aprovechando las características muy especializadas de las huellas de ejecución (los eventos llegan casi ordenados temporalmente y nunca son modificados una vez registrados), más el hecho de que todas las acciones de navegación requeridas por un depurador omnisciente pueden ser calculadas usando simples consultas de filtrado, pudimos diseñar un sistema particularmente escalable: la base de datos de eventos se puede paralelizar, y en nuestros experimentos usando un cluster de 10 máquinas, pudo resistir por largos ratos un flujo de entrada de un medio millón de eventos por segundo.[1] Sin embargo, el mismo hecho de capturar la huella de ejecución de un programa tiene un impacto significativo sobre éste: se puede observar una pérdida de eficiencia de hasta 80 veces en el peor caso (es decir, un programa totalmente instrumentado y haciendo uso intensivo de CPU), aunque es posible reducir netamente ese impacto excluyendo partes del programa del proceso de instrumentación (por ejemplo, las clases del JDK).

En nuestra propia experiencia, una configuración de una sola máquina es suficiente para huellas relativamente pequeñas (más o menos 10 millones de eventos) y una configuración con dos máquinas (es decir, con una máquina dedicada a la base de datos además de la máquina de desarrollo) puede aguantar cómodamente huellas de 150 millones de eventos, lo que resulta suficiente para depurar, por ejemplo, la base de datos de eventos de TOD. Para huellas más grandes, organizaciones que se lo pueden permitir, se beneficiarían de una configuración más poderosa.

Depurando con TOD

TOD permite navegar temporalmente, con la ejecución paso a paso simulada hacia adelante y atrás, y hace posible navegar causalmente gracias a un vínculo desplegado al lado del valor de las variables inspeccionadas, lo que permite saltar directamente al evento que asignó a una variable su valor actual. A continuación

[1] El lector interesado se puede referir a nuestro trabajo previo para obtener más detalles [5].

describimos una sesión de depuración haciendo uso de esta funcionalidad (Figura 3a).

Después de correr el programa erróneo con el botón de arranque de TOD (1), podemos fácilmente localizar el evento correspondiente a la excepción en la huella de ejecución. Una vez que este evento se encuentra seleccionado en la vista principal de flujo de control (2), la línea de código respectiva resalta automáticamente (3). En ese momento nos damos cuenta de que el campo thumbnail del objeto ThumbnailPanel actual tiene un valor nulo (4), lo que causó la excepción. Haciendo clic en el vínculo *why?* (4) llegamos inmediatamente no sólo a la línea de código fuente donde se asigna el valor al campo (5), sino que también al evento preciso que causó esa asignación en particular (6). Nótese que la asignación ocurrió en un hilo de ejecución distinto en que ocurrió la excepción (7). Al inspeccionar el estado del programa en el momento de la asignación, uno se da cuenta que éste intentó crear una miniatura de un archivo .sh, lo cual falló.

En este ejemplo simple, TOD permitió saltar en algunos pocos pasos directamente desde el síntoma del error (la excepción) a su causa (el manejo erróneo de archivos que no son imágenes).

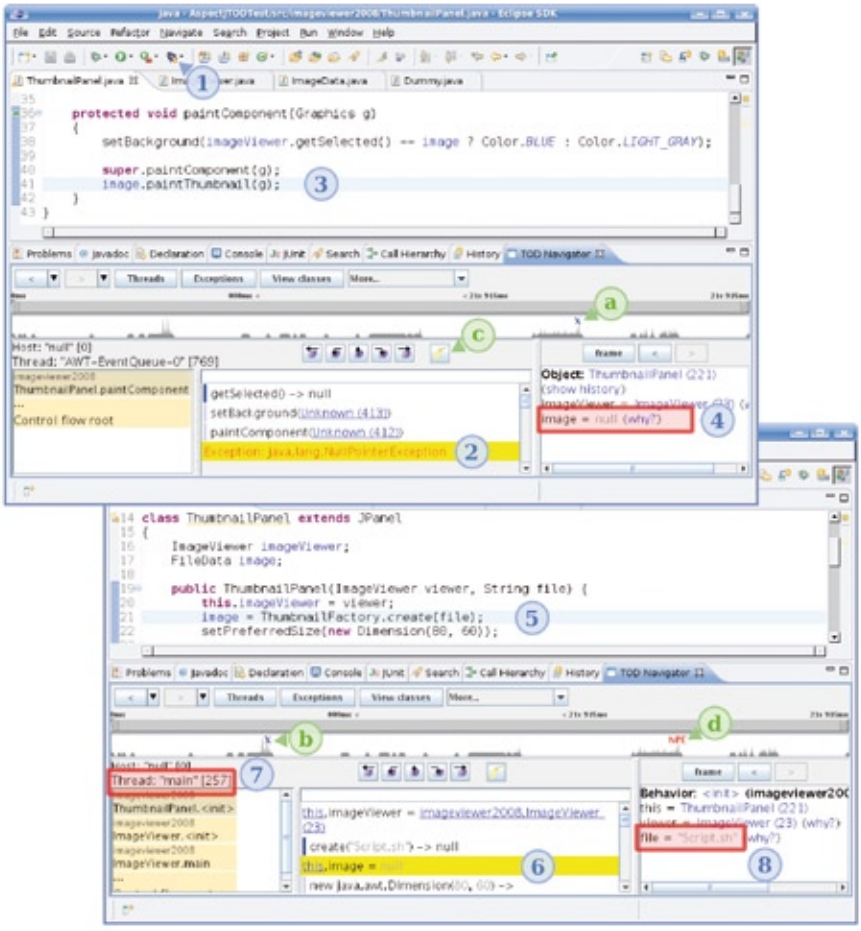
La misma búsqueda con un depurador basado en breakpoints hubiera sido más tediosa porque hay potencialmente muchos lugares en el programa donde el campo thumbnail es asignado, aparte del constructor, y hubiera sido necesario examinar paso a paso la ejecución de código sobre instancias válidas de ThumbnailPanel.

Un ejemplo de juguete como éste no muestra todo el potencial de TOD. Sería demasiado largo exponer aquí en detalles, pero podemos mencionar que hemos logrado usar TOD para solucionar rápidamente problemas difíciles, como errores en la base de datos de TOD, así como para entender problemas que encontramos en nuestro uso de programas altamente complejos como el compilador de AspectJ abc[2].

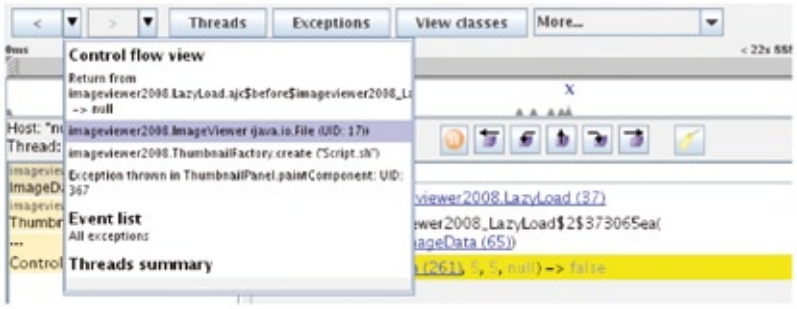
Usando marcadores para no perderse

Dada la enorme cantidad de eventos registrados por TOD, es de suma importancia ayudar al usuario a no perderse navegando en las huellas de ejecución. Para evitar eso, TOD deja al usuario marcar eventos y objetos, y permite acceder con rapidez a ubicaciones previamente visitadas.

Los eventos marcados están desplegados en una línea de tiempo arriba de la vista principal de TOD. El evento seleccionado en la vista principal también está indicado en la línea de tiempo, de tal manera que



(a) Buscando la causa raíz de un error de tipo NullPointerException usando el vínculo *why?*



(b) Historial de navegación.

Fig. 3 Interfaz de usuario de TOD.

[2] http://abc.comlab.ox.ac.uk/introduction

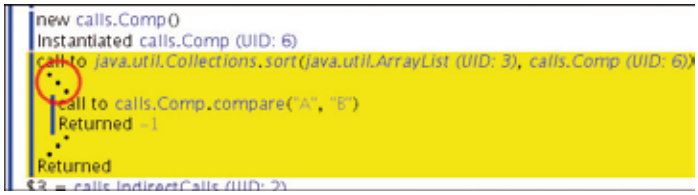


Fig. 4 TOD muestra la información que falta en una reconstitución de flujo de control.

el usuario pueda ubicarse rápidamente en relación a hitos conocidos. Esto, en particular es útil al usar el vínculo *why?* antes descrito, ya que éste puede causar saltos a eventos ocurridos lejos en el pasado, en contextos totalmente distintos.

Los globos verdes en la Figura 3a ilustran el proceso de marcación de eventos. La posición del evento actual está siempre marcada en la línea de tiempo (a, b). Cuando el usuario piensa que el evento actual es un hito importante, o un punto de partida para explorar varios recorridos de la huella de ejecución, puede marcarlo pulsando el botón de marcado (c), que también da la posibilidad de elegir un nombre y color para el evento (d).

Los objetos también se pueden marcar: es posible asignar un nombre y/o color a un objeto en particular, de tal manera que siempre aparecerá con esas características cuando esté desplegado.

Esto hace posible, por ejemplo, marcar un objeto involucrado en un error de tal manera que usos previos de este objeto sean destacados durante la navegación.

Además de los marcadores, la interfaz de usuario provee botones para navegar hacia delante y atrás, muy parecidos a los de un navegador web (Figura 3b), que permiten acceder rápidamente a todo el historial de navegación.

Soporte para huellas parciales

Si bien TOD está diseñado para aguantar huellas de ejecución muy grandes, no siempre es práctico registrar absolutamente todos los eventos: el impacto causado por la captura de huella sobre el programa depurado es considerable, así como lo son los requisitos de almacenamiento.

Para aliviar esta situación, uno puede aprovechar el hecho de que sólo algunas partes del programa son interesantes, capturando *huellas parciales* [5].

Con TOD, las huellas parciales se obtienen usando una mezcla de scoping *estático y dinámico*. El scoping estático consiste en seleccionar las clases que deben o no generar eventos. El scoping dinámico consiste en activar o desactivar la captura en tiempo de ejecución, ya sea con una simple API que se puede usar directamente desde el programa depurado, o con un botón en la interfaz del depurador. El scoping dinámico es particularmente útil cuando un error se produce sólo después de un largo tiempo de ejecución, o bajo condiciones dinámicas particulares. Por ejemplo, en una aplicación web, puede resultar interesante limitar la captura de huella al procesamiento de una consulta HTTP en particular.

El inconveniente de las huellas parciales es que son, justamente, parciales. Por lo tanto, algunas partes del historial del programa depurado no se pueden reconstituir. Para permitir al desarrollador razonar de forma correcta sobre la información disponible, TOD hace sistemáticamente *explícita* la información faltante. Por ejemplo, en la Figura 4, los puntitos indican que la información de flujo de control es incompleta: entre la ejecución de `sort` y la de `compare`, ocurrieron operaciones no registradas porque el método `Collections.sort` del JDK estaba fuera del scope estático.

En la práctica, los beneficios de las huellas parciales superan sus inconvenientes. En nuestra experiencia, combinar scoping estático y dinámico ha sido indispensable para depurar programas que hacen uso intensivo de la CPU por largos tiempos, tal como la base de datos de TOD.

ESTADO ACTUAL DEL ARTE

La mayoría de los ambientes de desarrollo modernos proveen un depurador basado en breakpoints. Todos tienen más o menos las mismas capacidades: breakpoints normales o condicionales, ejecución paso a paso hacia adelante, inspección del marco de activación actual y de los objetos alcanzables desde él. Pero también existen algunos depuradores omniscientes disponibles hoy. Los describimos brevemente a continuación.

Depuradores omniscientes para Java

ODB [3] es uno de los primeros depuradores omniscientes para Java. Al igual que TOD, obtiene huellas de ejecución instrumentando las clases a medida que están cargadas por la JVM; sin embargo, ODB almacena la huella de ejecución dentro de la JVM depurada, lo que causa algunos problemas: el tamaño de la huella está limitado por la memoria heap disponible, y referencias a objetos que ya no están en uso están conservadas, impidiendo la recolección de basura. Un aspecto único de ODB es su capacidad para reanudar la ejecución del programa desde cualquier punto en el tiempo con un estado alterado.

Java Whyline [6] deja el usuario seleccionar preguntas sobre por qué cierto comportamiento *ocurrió* o *no ocurrió*. Estas preguntas están generadas en forma automática usando una combinación de análisis estático y dinámico, y pueden abarcar no solamente el estado interno del programa (por ejemplo, "¿por qué la variable x tiene el valor y?"), sino también su salida textual y gráfica, hasta el nivel de píxeles individuales. A pesar de que Whyline puede

aguantar huellas relativamente grandes (por ejemplo, 35 millones de eventos), su escalabilidad es limitada por el hecho de que el análisis se hace en memoria.

JIVE [7] es un *ambiente de visualización* interactivo para programas Java. Provee diagramas de secuencia parecidos a los de UML, pero extendidos con información acerca del llamado de método actual. El nivel de detalle de los diagramas puede ser reducido para poder desplegar más información en pantalla, pero no es claro que este mecanismo pueda escalar a más de unos centenares de elementos. JIVE soporta ejecución paso a paso hacia adelante y atrás, pero no navegación causal rápida. La huella de ejecución es capturada con JPDA, la interfaz de depuración de la JVM, y procesada en memoria, lo que limita la escalabilidad del sistema.

Otras plataformas

Lisp: En el 1984, ZStep [8] proveía un simulador de ejecución paso a paso para Lisp: permitía ir paso a paso hacia adelante y atrás, así como ver el resultado de la evaluación de las expresiones en paralelo al código fuente correspondiente. Su seguidor, ZStep95 [9], agregó la posibilidad de relacionar las salidas gráficas del programa al evento que la causó, y también proveía controles similares a una grabadora de cinta para facilitar la navegación. Sin embargo estos sistemas no manejaban efectos de borde, vínculos causales (excepto para las salidas gráficas), o problemas de escalabilidad.

Nativos:

TimeMachine, de Green Hills Software[3], es un depurador omnisciente para sistemas embedidos (PowerPC, ARM y arquitecturas similares). En algunas plataformas, una sonda de hardware permite capturar huellas de ejecución sin tener ningún impacto sobre el programa depurado; para las otras plataformas se usa la tradicional instrumentación de código. Además de las funcionalidades habituales de los depuradores omniscientes, TimeMachine se puede usar como herramienta de profiling.

UndoDB, de Undo Ltd[4], es un depurador omnisciente para programas Linux x86 nativos. Al contrario de la mayoría de las otras herramientas presentadas aquí, es basado en un mecanismo de instantáneo/reejecución: se obtiene periódicamente un *instantáneo* (o *snapshot*) de la memoria del proceso, y se usa una técnica de reejecución para reconstituir el estado del programa entre los instantáneos. El uso de este mecanismo resulta en un impacto reducido sobre el programa depurado, pero no permite navegación causal.

	Platform	*Mechanism*	*Storage media*	*History size*	*Runtime overhead*	*Partial traces*	*Causal nav.*	*High-level overviews*	*IDE integration*
TimeMachine	Embedded	?	RAM/ Probe	1e9	Soft.: ? Hard.: none	?	?	✔	Part of Green Hill's MULTI IDE
UndoDB	Linux	Checkpoint replay	RAM	Not applicable	7x	Not applicable	✘	✘	Wrapper for gdb
Chronicle	Linux	Event log	Disk	1e9	300x	?	✔	✘	Plugin for Eclipse CDT
[Lienhard]	Squeak	Event log	RAM	1e5	6x	Events on unreachable objects are discarded	✘	✘	Integrates into the platform
Unstuck	Squeak	Event log	RAM	1e5	250x	Lexical scoping	✘	✘	Integrates into the platform
Whyline	Java	Event log	Disk	1e7	252x/20x	Lexical scoping	✔	✘	No
JIVE	Java	Event log	RAM	?	?	Lexical scoping	✘	✔	Plugin for Eclipse JDT
ODB	Java	Event log	RAM	1e6	95x/37x	Lexical scoping	✔	✘	Limited Eclipse integration
TOD	Java	Event log	Disk	1e9	83x/28x	Lexical & dynamic scoping Missing info explicit in GUI	✔	✔	Plugin for Eclipse JDT/AJDT

Fig. 5

La columna *History size* indica el orden de magnitud de la cantidad de eventos que se pueden razonablemente almacenar y procesar. *Runtime overhead* da una idea del impacto sobre el programa depurado. Para los sistemas en los cuales hicimos nuestros propios experimentos, aparecen dos cifras (X/Y) donde X es el impacto en el peor caso (es decir, un programa enteramente instrumentado que usa intensivamente la CPU), e Y corresponde a una situación más típica (en nuestro caso, una ejecución del limpiador de código HTML jTidy). Para los demás sistemas, la única cifra es la indicada por el propio autor del sistema. En la columna *Partial traces, lexical scoping* significa que es posible seleccionar las clases o paquetes que se instrumentan, y *dynamic scoping* significa que se puede activar o desactivar la captura en tiempo de ejecución. *Causal navigation* indica si el depurador permite navegar directamente al evento que asignó su valor actual a una variable. *High-level overviews* indica si el sistema puede proveer vistas resumidas del programa depurado.

[3] http://www.ghs.com/products/timemachine.html
[4] http://undo-software.com/

Chronicle[5] es un depurador omnisciente open-source para programas Linux x86 nativos. Su arquitectura es similar a la de TOD: los binarios son instrumentados de tal manera que envían la huella de ejecución a una base de datos externa, almacenada en disco. Una característica clave de Chronicle es la compresión e indexación agresiva de los eventos, lo que permite registrar huellas muy grandes y procesar consultas eficientemente.

Smalltalk:

Unstuck [10] es un depurador omnisciente para Smalltalk, muy similar a ODB en su arquitectura y operación. Lienhard *et al.* [11] proponen otro depurador omnisciente para Smalltalk que trata el problema de escalabilidad usando huellas parciales, pero en una manera muy distinta a la de TOD. Ellos postulan que la información acerca de objetos que ya no están alcanzables en un cierto momento (es decir, objetos que pueden ser recolectados por el recolector de basura) puede ser descartada. Si bien descartar esa información mejora mucho la eficiencia del sistema, pensamos que la causa raíz de un error puede haber ocurrido en el contexto de objetos que han sido descartados mucho antes de que los síntomas del error se manifiesten, lo que vuelve este enfoque inoperativo en algunos casos.

Resumen

La figura 5 resume las características de las herramientas antes descritas. A continuación mencionamos las características de TOD que, en nuestra opinión, hacen de él una alternativa competitiva:

- La base de datos escalable permite registrar y consultar eventos en forma rápida. Además, puede ser distribuida sobre un cluster de máquinas para aumentar aún más su eficiencia. Eso hace de TOD el depurador omnisciente más escalable para la plataforma Java.
- El soporte para huellas parciales aumenta de sobremanera la aplicabilidad de TOD porque ofrece una manera expresiva de especificar una captura de huella selectiva, y porque reporta de manera adecuada información faltante. Pensamos que TOD provee el soporte más extensivo para huellas parciales.
- La interactividad de la interfaz, obtenida gracias al procesamiento rápido de consultas, permite navegar en forma interactiva en huellas de ejecución muy grandes.
- Las metáforas de interacción especializadas, como el vínculo *why?*, los marcadores y las líneas de tiempo, permiten una navegación y comprensión de programas eficiente.
- La integración con Eclipse permite integrar fácilmente TOD en el proceso de desarrollo.[6] Por otro lado, algunas de las funcionalidades otorgadas por diversos sistemas hacen falta en TOD:
- Whyline permite formular preguntas negativas como *"¿por qué el método foo no se ejecutó?"*; ese tipo de preguntas es recurrente en el proceso de depuración.
- Whyline permite relacionar las *salidas textuales y gráficas* del programa con el evento que las causó. El soporte para salidas textuales en TOD está considerado, pero el soporte para las salidas gráficas requeriría un trabajo considerable.
- TOD tiene un *impacto sobre el programa depurado* similar al de ODB y de Whyline, y a la vez provee una escalabilidad mucho mayor. Sin embargo, sistemas como UndoDB y el de Lienhard tienen un impacto mucho menor, mientras TimeMachine puede no tener ningún impacto al usar sondas de hardware. Nos esforzamos por reducir el impacto de TOD usando análisis estáticos para limitar la cantidad de información redundante capturada.
- JIVE provee *visualizaciones del grafo de objetos,* lo que puede ser muy útil para la comprensión de programas. Aunque TOD soporta formateadores personalizados, estos son solamente textuales.

PERSPECTIVAS

Mientras la depuración omnisciente parece estar atrayendo poco a poco más la atención de los industriales, nuevos desafíos están apareciendo con la emergencia de nuevos lenguajes y paradigmas de programación. Ya podemos identificar tres áreas mayores en las cuales el desarrollo de sistemas de depuración omnisciente prácticos es crucial:

- Los lenguajes dinámicos (como Python, Ruby...) se están haciendo cada vez

[5] http://code.google.com/p/chronicle-recorder/
[6] La integración con NetBeans e IntelliJ está en curso.

más populares. Mejorar el soporte de depuración ayudaría a aliviar, al menos hasta cierto grado, la ausencia de verificación de tipos estática.

- Desarrollar sistemas concurrentes y distribuidos es notoriamente difícil, en particular porque errores pueden ser difíciles de reproducir. Ser capaz de registrar automáticamente historiales de ejecución para luego navegar en ellos es entonces de gran importancia.
- Porque agrega más lugares donde ocurre el enlace tardío, la Programación Orientada a Aspectos (AOP) [12] hace más difícil para los programadores la reconstrucción mental del flujo de ejecución de los programas. Se requieren herramientas de desarrollo adecuadas, y en particular depuradores, para soportar AOP.

Ya estamos explorando como la depuración omnisciente puede proveer un soporte adecuado para AOP [13]. También estamos desarrollando una versión de TOD para Python. La programación concurrente y distribuida también está en nuestra agenda de investigación.

Dado lo eficiente que es la depuración omnisciente para la comprensión de programas, mejora de sobremanera el proceso de desarrollo de software. Es entonces sumamente importante dedicar esfuerzos para hacerla práctica y aplicable a la mayor cantidad de situaciones posibles, resolviendo los distintos desafíos para su adopción. BITS

Disponibilidad. TOD está disponible en http://pleiad.dcc.uchile.cl/tod/

REFERENCES

[1] National Institute of Standards and Technologies, "Software errors cost U.S. economy $59.5 billion annually," June 2002. http://www.nist.gov/public affairs/releases/n02-10.htm.

[2] M. Eisenstadt, "My hairiest bug war stories," *Commun. ACM*, vol. 40, no. 4, pp. 30–37, 1997.

[3] B. Lewis, "Debugging backwards in time," in *Proceedings of the Fifth International Workshop on Automated Debugging (AADEBUG 2003)* (M. Ronsse and K. D. Bosschere, eds.), (Ghent, Belgium), 2003.

[4] R. M. Balzer, "EXDAMS– extendable debugging and monitoring," in *Proceedings of the AFIPS Spring Joint Computer Conference*, pp. 567–580, 1969.

[5] G. Pothier, É. Tanter, and J. Piquer, "Scalable omniscient debugging," in *Proceedings of the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA 2007)*, (Montreal, Canada), pp. 535–552, Oct. 2007. , 42(10).

[6] A. J. Ko and B. A. Myers, "Debugging reinvented: Asking and answering why and why not questions about program behavior," in ICSE 2008: *Proceedings of the International Conference on Software Engineering*, 2008.

[7] P. Gestwicki and B. Jayaraman, "Methodology and architecture of JIVE," in SoftVis '05: *Proceedings of the 2005 ACM symposium on Software visualization*, (New York, NY, USA), pp. 95–104, ACM, 2005.

[8] H. Lieberman, "Steps toward better debugging tools for lisp," in LFP '84: *Proceedings of the 1984 ACM Symposium on LISP and functional programming*, (New York, NY, USA), pp. 247–255, ACM, 1984.

[9] H. Lieberman and C. Fry, "ZStep 95: A reversible, animated source code stepper," in *Software Visualization — Programming as a Multimedia Experience* (J. Stasko, J. Domingue, M. H. Brown, and B. A. Price, eds.), (Cambridge, MA-London), pp. 277–292, The MIT Press, 1998.

[10] C. Hofer, M. Denker, and S. Ducasse, "Implementing a backward-in-time debugger," in *Proceedings of NODe'06*, vol. P-88, pp. 17–32, Lecture Notes in Informatics, 2006.

[11] A. Lienhard, T. G˜A R rba, and O.N˜ ierstrasz, "Practical object-oriented back-in-time debugging," in *Proceedings of ECOOP'08: European Conference on Object-Oriented Programming (to appear)*, 2008.

[12] T. Elrad, R. E. Filman, and A. Bader, "Aspect-oriented programming," *Communications ACM*, vol. 44, Oct. 2001.

[13] G. Pothier and É. Tanter, "Extending omniscient debugging to support aspect-oriented programming," in *Proceedings of the 23rd ACM Symposium on Applied Computing (SAC 2008)*, vol. 1, (Fortaleza, Cear´a, Brazil), pp. 266–270, Mar. 2008.

Nota y Agradecimientos. Este artículo es una traducción y leve adaptación de nuestro artículo "Back to the Future: Omniscient Debugging" aceptado para publicación en el journal IEEE Software.

Agradecemos a Greg Law de Undo Ltd. por proveer información técnica sobre UndoDB, así como a Alexandre Bergel, Johan Fabry, Adrian Lienhard, Olivier Motelet y los revisores anónimos de IEEE Software por sus valiosos comentarios.

Antes del Internet
(Una visión personal)

Centro de computación (C.E.C.), FCFM. **Fotografía: Gastón Carreño.**

José Miguel Piquer
Profesor Asociado, DCC, Universidad de Chile. Doctor en Computación, École Polytechnique de Paris. Director Técnico de NIC Chile y Director de NIC Labs.
jpiquer@dcc.uchile.cl

EL COMIENZO DE LA HISTORIA

Yo vivía corriendo por el pasillo del DCC, entre mi oficina y la sala de máquinas, para llegar a tiempo a mover un switch en el computador después de dar un comando en la consola que estaba en mi oficina, única forma de rebootearlo. En esos tiempos, la sala de máquinas del DCC era una especie de closet con dos computadores NCR Tower y un aire acondicionado de ventana (que daba al pasillo) que funcionaba a veces. A esos computadores les tomé un cariño infinito, porque (además de ser el primer Unix que conocí) a veces se rehusaban encender (sobre todo en días calurosos) y había que abrirlos, sacarles el disco duro principal y, con él en mi mano, hacerlos partir. La sensación de revivirlos, con un disco emitiendo un ligero zumbido en la mano, eso es algo que lo marca a uno para siempre...

Corría el año 1985, el DCC ocupaba un pedazo del primer piso de Blanco Encalada 2120 y una de nuestras prioridades era lograr que el resto de la Facultad dejara de confundirnos con el CEC (Centro de Computación) que ocupaba todo el resto del primer piso, ¡además del segundo!, y era tremendamente famoso. De los profesores que están hoy, sólo José Pino y Patricio Poblete ya eran profesores del DCC. Jorge Olivos (actual director del CEC) era el director del DCC, yo era todavía un estudiante de Magíster en Computación, junto con Ricardo Baeza-Yates, Nancy Hitschfeld y Nelson Baloian; y trabajaba en el DCC como administrador de sistemas.

Edgardo Krell (actualmente Director de Operaciones y Sistemas de NIC Chile) era profesor en la USACH en ese entonces y junto con Sergio Mujica (ex director de la Escuela de Ingeniería Informática de la Universidad Diego Portales) y nosotros,

configuramos un par de modems de 300 bps conectados a la línea telefónica y al NCR Tower, uno idéntico al nuestro, que tenían en el Departamento de Ingeniería Informática de allá. Ambos computadores fueron una donación de la empresa NCR; un hito fundamental pues nos dieron acceso por primera vez a computadores corriendo Unix. El principal problema era encontrar una línea telefónica para usar, porque necesitábamos un teléfono directo y eran muy escasos: sólo los directores de departamento tenían uno y, por supuesto, no lo iban a regalar para que lo usaran los computadores.

En el DCC conseguimos tener una línea propia, todo un lujo, pero en la USACH hubo que instalar un switch manual que Edgardo se encargaba de mover a la posición "computador" a la hora de almuerzo y cuando se acababa la jornada. Más importante aun era volverlo a la posición "teléfono" cuando aparecía el director.

Con esa configuración, logramos conectar a los computadores entre sí, y utilizar el protocolo UUCP para intercambiar archivos. Con eso funcionando, llegó la hora de probar el mail, para lo cual Patricio Poblete envió a Sergio Mujica el famoso mail que decía:

```
si este mail te llega, abramos
una botella de champaña
```

(el mail aún no soportaba acentos ni eñes). El mail llegó y definió un hito histórico del que teníamos muy poca conciencia. Ni lo guardamos, ni tampoco abrimos ninguna botella.

Poco tiempo después armamos una conexión parecida con la Universidad Católica. Con Ignacio Casas (actualmente director del SECICO) y el apoyo de Edgardo y Sergio (todos ellos se conocían bien) se instaló Ultrix en la VAX de allá.

Bastante rápido descubrimos que tener mail entre pocas personas no tenía mucha gracia y que necesitábamos conectarnos con el resto del mundo. Fue así como instalamos una primera conexión internacional con el INRIA en Francia usando X.25 en vez del teléfono. Como era una red de datos internacional, pensamos que iba a salir más barato, pero en realidad no resultó como esperábamos. Aunque fue espectacular el poder recibir y enviar mail a todo el mundo, el costo resultó ser muy alto, porque se tarificaba por byte transmitido, y cada vez que fallaba un envío se pagaba cada reintento. En un contexto universitario donde no había dinero ni para comprar tiza, el costo se nos volvió rápidamente inmanejable.

Después de unas semanas, descubrimos que nuestros mails no lograban pasar más allá de Europa. Al averiguar un poco más, el administrador de sistemas en Francia (Yves Devillers) descubrió que nos habían bloqueado en Holanda, donde estaba el gateway principal entre Europa y América: EuNet. El administrador de EuNet, Piet Berteema, resultó un holandés durísimo y hubo una discusión muy divertida entre Yves y Piet ya que uno sostenía que Chile estaba más cerca de Europa (culturalmente) y el otro de Estados Unidos (geográficamente). En definitiva, ganó Piet, pero nos generaron un contacto en Estados Unidos: Rick Adams, quien trabajaba en el Center for Seismic Studies, que operaba una máquina llamada "seismo", que se había transformado en un punto de interconexión fundamental de la red UUCP en el mundo.

Esa conexión con Seismo la comenzamos con X.25 también, pero rápidamente la reemplazamos por una conexión telefónica larga distancia usando modems Telebit Trailblazers, recomendados por Rick y traídos directamente de Estados Unidos. Nunca habíamos visto un modem tan rápido en nuestra vida: ¡9 Kbps! (en teoría podía llegar a 19.2 kbps, pero no con la calidad de nuestra telefonía internacional).

Esto permitió tener mail internacional en forma habitual y, para usar el teléfono en los horarios acordados, nos permitía enviar un mail en la mañana, el que se transmitía al almuerzo y, si se respondía inmediatamente, la respuesta nos podía llegar durante la noche y nosotros leerla a la mañana siguiente. Una ida y vuelta en 24 horas. Estábamos realmente impresionados.

Patricio Poblete también logró que Rick nos enviara las news completas en cintas magnéticas, lo que nos hizo entrar al mundo de Usenet muy al comienzo. Nuestras opiniones se enviaban por la conexión telefónica, y recibíamos los foros por cinta una vez a la semana.

En 1986 terminé mi Magister en Computación y, de inmediato, me contrataron como profesor jornada completa. En ese tiempo, Ricardo Baeza-Yates (ya contratado) se fue a doctorar a la Universidad de Waterloo en Canadá y contrataron a varios sobrevivientes que aún andan por el DCC: Nancy Hitschfeld, nuestra actual directora, Nelson Baloian y Luis Mateu. Recuerdo que un profesor nuevo del DCC tenía un Mac y vio que se publicaban aplicaciones en la red para bajarlas y usarlas en Mac, por lo que le pidió a Ricardo que mirara cosas interesantes y se las enviara por mail. Ricardo, fiel a su estilo, copió todas las aplicaciones Mac que estaban publicadas y se las envió como respuesta. Nunca llegaron, pero estuvieron una semana tratando de transmitirse a través de nuestros modems Trailblazers, gastando horas de telefonía larga distancia sin terminar nunca su esfuerzo. Cuando descubrimos el problema, pedimos a Rick que borrara ese mail de la cola de envíos, pero ¡a esa altura ya debíamos casi tanto dinero como el presupuesto de operación anual del DCC!

Bastante rápido descubrimos que tener mail entre pocas personas no tenía mucha gracia y que necesitábamos conectarnos con el resto del mundo.

EL DOMINIO .CL

En 1987, justo antes de partir a mi doctorado a École Polytechnique de París, nos tocó enfrentar un cambio fundamental en el sistema de mail: UUCP había decidido adoptar la notación de mail de Internet, con nombres de dominio. Hasta ese día, mi mail era:

```
...!seismo!uchdcc!jpiquer
```

Donde los puntos suspensivos había que reemplazarlos por el camino que el mail debía seguir para llegar a seismo (¡todos sabían como llegar allí!).

Así fue que Rick nos pidió que registráramos un nombre de dominio bajo .CL, que era el nombre que tenía asignado Chile para esto. Eso iba a permitir que todos nuestros mails fueran como los conocemos hoy en día, eliminando la secuencia de caminos a seguir. Nosotros le contestamos de vuelta que estaba bien, pero que nos dijera quién administraba .CL para pedirle un nombre para nosotros. Rick le envió la pregunta a Jon Postel, quien administraba todos los nombres de primer nivel como la IANA (Internet Assigned Numbers Authority) y su respuesta fue: nadie ha pedido administrarlo aún, ¿por qué no se hacen cargo ustedes?

Nos enviaron un pequeño formulario por mail que llenamos rápidamente. Recuerdo que Jorge Olivos quedó como Contacto Administrativo y yo como Contacto Técnico. El servidor de nombres, que debía correr con conexión dedicada a Internet, lo corrió Rick en su servidor: UUNET, que ya operaba en la empresa que había creado para desarrollar la conectividad de redes privadas a Internet y sacar el servicio del Instituto de Sismología. UUNET llegó a ser el ISP más grande del mundo y fue comprada finalmente por Worldcom.

En el formulario había que marcar qué tipo de servidor de nombres uno ejecutaba y la alternativa estándar era "BIND". Yo no entendía de qué estaban hablando y les pregunté ¿qué es eso? Nunca pensé que iba a vivir los próximos 20 años dependiendo de BIND...

Con esos simples pasos, quedamos oficialmente a cargo del dominio .CL, donde inicialmente lo único que existía era un registro MX (Mail Exchanger) que decía que todos los computadores terminados en .CL eran atendidos por nuestro viejo uchdcc, que ahora se llamó uchdcc.cl para marcar la diferencia. Desde 1987 hasta 1993 operamos de esa forma, con el servidor primario en UUNET y haciendo las modificaciones allá vía mail.

Una de las primeras decisiones que tuvimos que tomar fue si íbamos a seguir el modelo de subdividir el dominio en .com.cl, .edu.cl, etc o íbamos a permitir que las instituciones inscribieran nombres directamente en el primer nivel. Recuerdo un almuerzo donde conversamos eso, y yo planteé que tal vez no era mala idea subdividir, porque así el dominio .CL no quedaba tan grande. Patricio Poblete, con su clásico estilo de matemático, me argumentó que dividir tamaño del dominio por 4 (o cualquier constante, en realidad) no cambiaba el orden del problema y eso terminó de convencerme. En todo caso creo que el principal argumento a favor de no tener subdivisiones fue que nuestro nombre de dominio en el DCC iba a quedar demasiado largo: anakena.dcc.uchile.edu.cl parecía absurdo.

LA CONEXIÓN A INTERNET

Justo a mi vuelta a Chile, en 1991, se estaba gestando el proyecto de conectar a las universidades chilenas a Internet. Como yo venía de ser usuario de Internet en Francia, me tomaron como un experto en Internet, por lo que asesoré la puesta en marcha del proyecto desde la Universidad de Chile y aprendí mucho en el camino.

Fue en ese período que descubrí lo difícil que puede ser la política en torno a los temas técnicos. Con el retorno a la democracia, comenzó una lucha de poder muy fuerte entre la Universidad de Chile y la Universidad Católica, tratando ambas de liderar el proyecto. Siempre había existido un liderazgo claro de la Universidad de Chile en los temas de desarrollo de redes: la red UUCP todavía funcionaba y su nodo central estaba en el DCC y la red BITNET (que conectaba a los mainframes IBM en una red de correo electrónico) tenía su nodo central en el CEC. En torno a esas iniciativas, se buscaba crear un proyecto nacional de infraestructura de conectividad entre universidades, que operara una troncal Internet, equivalente a lo que en Estados Unidos era la NSFnet. Durante ese año participamos en el debate siempre empujando una solución integradora, donde fueran todos los actores, hablando incluso con el rector Lavados, con D'Etigny como presidente de CONICYT y Florencio Utreras como director del CEC. En definitiva, a fin de 1991 había resultado imposible llegar a un proyecto único, y la Universidad de Chile lideró el proyecto REUNA que integraba a la mayoría de las Universidades del Consejo de Rectores y la Universidad Católica lideró el proyecto UNIRED, sumando a la USACH y a la Universidad Católica del Norte. Comenzó entonces una batalla épica entre los dos proyectos por lograr su conexión a la NSFnet primero, mientras Steve Goldstein, de la NSF, nos repetía majaderamente: "One country, one link". Entiendo que tenemos el triste record de ser el primer país en lograr que no se respetara esa regla y, enero de 1992, ambos proyectos quedaron conectados con su respectivo enlace satelital a 64 Kbps (considerado "banda ancha" en esa época). Aunque no anotamos la fecha, parece ser que el jueves 9 de enero de 1992 logramos que el primer paquete IP (un ping, por supuesto) transitara entre los routers Proteon del CEC (en el segundo piso de Blanco Encalada 2120) y de la Universidad de Maryland, mientras saltábamos de alegría con el gringo en el teléfono. Recuerdo que lo último que nos dijo el gringo era que consideráramos comprar unos routers mucho más baratos y eficientes que habían salido recién de una empresa que nadie conocía que se llamaba Cisco.

La fecha parece ser correcta, porque uno de los participantes del hecho recuerda que nos invitó a tomar una cerveza para celebrar, y yo les dije que tenía que salir corriendo porque ya estaba atrasado para irme a celebrar mi aniversario de matrimonio. Curiosamente, ese mes cumplí justo 10 años de casado.

Mirando en retrospectiva, es difícil evaluar si esta batalla entre las universidades fue buena o mala. Es verdad que generó una motivación y un apuro por lograr la meta que no habría existido en un ambiente de cooperación. Pero también generó una pésima solución de conectividad, que hizo por muchos años que la mayoría del tráfico entre proveedores en Chile pasara por Estados Unidos. Lo peor es que generó un mal modelo, que CTC y Entel copiaron después, lo que perpetuó esta mala conectividad hasta el año 2000.

Lo que sí fue positivo, al menos para nosotros, fue que mantuvimos la administración del dominio .CL gracias a esta disputa. En esos años, si hubiese habido un proyecto consensuado entre todos de hacer un gran administrador del Internet nacional, probablemente habríamos entregado el dominio, la red UUCP y todos los servicios que dábamos desde la Universidad de Chile a dicha entidad. Sin embargo, en el ambiente de conflicto que se vivió, resultaba sumamente irresponsable traspasar el dominio a alguna de las dos entidades creadas y, a pesar de la presión que ejerció REUNA para obtenerlo, terminamos quedándonos con él. Curiosamente, la Universidad Católica fue un firme partidario de que nosotros administráramos el dominio, porque estaban convencidos de que REUNA lo usaría en su contra si obtenía su administración.

LA PRIMERA PÁGINA WEB

Durante 1993, ya Internet comenzaba a transformarse en algo natural en el funcionamiento del DCC. El mail era instantáneo, las news llegaban a tiempo y podíamos transferir archivos con el mundo entero. REUNA comenzaba a operar como un consorcio independiente y se dedicaba a conectar a las universidades regionales en base a un proyecto FONDEF. Durante ese año, intercambiamos los roles de los servidores de .CL con UUNET, de modo que nosotros comenzamos a actuar como primario y ellos quedaron como secundario.

Yo dicté un curso tipo Taller de Sistemas ese año, donde los alumnos debían instalar servicios interesantes y hacer pruebas y demostraciones. En un pasillo del DCC (en ese entonces en el primer piso de Blanco Encalada) me crucé con uno de los alumnos, José (Pepe) Flores Peters. Ambos recordamos ese encuentro pero tenemos versiones sutilmente distintas. Yo recuerdo que él me planteó que había visto un servicio muy interesante, que permitía mostrar imágenes, texto y enlaces a otros servidores, usando Internet. Él recuerda que yo le dije eso mismo, y entonces tomó la idea de usarlo como su proyecto del curso. De lo que estoy convencido es que a mí no me entusiasmó mucho. Pensé que era absurdo mezclar todo en una interfaz común, con imágenes y cosas pesadas, que Internet no servía para eso.

En definitiva, el proyecto de Pepe se transformó en la primera página web de Chile y, algo que supimos mucho después, en la primera de latinoamérica. Además, a él le cambió la vida para siempre, porque se dedicó a esto: creó la primera empresa chilena que hacía páginas web (Tecnonáutica) y compitió varios años con la única otra empresa existente que había salido de la Universidad Católica y que, finalmente, fue comprada por Telefónica para morir dentro de su burocracia infernal.

CONCLUSIONES

¿Qué podríamos concluir de esta historia? Lo más triste fue ver cómo, a pesar de todos los esfuerzos que hicimos para evitarlo, la entrada a Internet fue peleada, dividida y mal organizada. En esa misma época llegaron dos chilenos al mismo tiempo a la cumbre del Everest y, en vez de abrazarse en celebración, se trenzaron a golpes discutiendo quién había sido el primero.

Pero tal vez lo más importante es descubrir que un pequeño grupo de gente, con mucho entusiasmo y pocos recursos, sí puede cambiar el mundo. Bueno, el mundo es mucho, pero cambiamos muchas cosas alrededor nuestro, influimos en muchas iniciativas y logramos que Chile, la Universidad de Chile y el DCC, entraran en muy buena forma en la historia del Internet.

> ...contamos con mucha suerte: estuvimos en el lugar adecuado en el momento adecuado.

Fotografía: Gastón Carreño.
C.E.C. FCFM.

Obviamente contamos con mucha suerte: estuvimos en el lugar adecuado en el momento adecuado. Pero creo que podemos sentirnos orgullosos de haber tomado esa oportunidad, haberle dedicado un enorme esfuerzo sin esperar nada a cambio, y haber logrado salir adelante exitosamente.

El gran desafío que nos queda es aprovechar hoy, de la mejor forma posible, el posicionamiento y liderazgo de esos años para potenciar aún más la solidez del DCC sólido, y contribuir con un país cada vez mejor. BITS

Contando Citas
en Artículos de Revistas y Conferencias

Mauricio Marín
Sociedad Chilena de Ciencia de la Computación. Yahoo! Research Latin America, Universidad de Chile. PhD en Computer Science, University of Oxford, UK.
mmarin@yahoo-inc.com

RESUMEN

En este trabajo se presentan estadísticas que muestran la tendencia observada en las ultimas décadas sobre el tipo de publicaciones realizadas en ciencia de la computación. Utilizando como métrica de calidad la cantidad de citas que reciben los artículos de los medios de publicación más difundidos en la disciplina, se compara el impacto de artículos publicados en revistas y conferencias.

Los resultados muestran que las publicaciones en conferencias pueden ser muy relevantes para determinadas áreas de la ciencia de la computación. También el impacto de estas publicaciones en el desarrollo de la disciplina ha ido creciendo en importancia en los últimos años. En varios casos el impacto es mayor que el de los artículos publicados en buenas revistas del área. En este artículo se describen los experimentos realizados para fundamentar estas tres afirmaciones.

Un artículo publicado en una conferencia o congreso es un documento completo de varias hojas con discusión y desarrollo similar al de un artículo de revista científica, el cual ha sido evaluado y seleccionado por un comité científico internacional, con tasas de aceptación bajo 30% y publicado por una de las casas editoriales reconocidas en la disciplina tales como IEEE-CS, ACM y LNCS de Springer. Gran parte de los artículos en revistas y conferencias en ciencia de la computación son indexados por medios alternativos a la indexación del \Web of Science" tales como DBLP, \ ACM Digital Library Portal" y CiteSeer^{x}. Los experimentos del presente artículo utilizan dichos sistemas de indexación como fuente de información para contabilizar citas por cada medio de publicación.

CITAS BIBLIOGRÁFICAS DEL "ACM COMPUTING SURVEYS"

El ACM Computing Surveys es una revista con volúmenes publicados desde el año 1969, en la cual los artículos tienen la forma de revisiones del estado del arte en tópicos específicos de ciencia de la computación. Generalmente se trata de tópicos que ya han sido investigados profundamente al momento de la publicación. Por lo tanto la lista de referencias bibliográficas de dichos artículos da cuenta de manera exhaustiva del estado del arte en el tema y, por supuesto, dichas referencias incluyen prioritariamente los trabajos que presentan las contribuciones más relevantes.

El sitio Web del ACM Computing Surveys muestra todos los volúmenes y números de esta revista a partir del año 1969. Por cada artículo se muestra la lista de referencias bibliográficas. Utilizando esa información calculamos la división entre el total de conferencias detectadas y el total de referencias bibliográficas de cada artículo, considerando la suma de conferencias y revistas.

También calculamos promedios anuales. Consideramos sólo los artículos del ACM Computing Surveys con más de 15 referencias para filtrar artículos tales como Cartas al Editor. El sitio Web también muestra en la lista de referencias bibliográficas de cada artículo las publicaciones que están indexadas en el Portal de la ACM Digital Library. Esto lo hace mediante un enlace al artículo registrado en el Portal, el cual indexa tanto artículos de conferencias como artículos de revistas e indica el tipo de publicación, el que está claramente diferenciado en el contenido del respectivo enlace. Con esto podemos calcular de manera exacta la proporción entre artículos de conferencias y de revistas citados en las listas de referencias bibliográficas.

En la Figura 1 se muestran resultados que abarcan las listas de referencias bibliográficas de artículos que fueron publicados entre 1969 y 2007. Los resultados indican una clara tendencia al alza en la proporción de artículos de conferencias que son citados. Es interesante ver que la Figura además de mostrar los promedios anuales (curva etiquetada con un círculo negro), también muestra la proporción por cada artículo individual (valores indicados con líneas verticales). Se observa que dependiendo del tópico del artículo, la proporción de estos en conferencias puede ser muy dominante en la lista de referencias (más de un 60%). Para otros tópicos, la contribución de los artículos de conferencias puede ser considerada irrelevante (menos de 30%).

FACTORES DE IMPACTO Y CITAS SEGÚN CITESEER^X

El CiteSeer^X (http://citeseerx.ist.psu.edu) es un indexador y máquina de búsqueda especializado en publicaciones en ciencia de la computación. Contiene información estadística desde 1993 a la fecha sobre datos tales como número de citas y utiliza la misma fórmula del ISI para calcular el factor de impacto de revistas y conferencias. Este factor, para un año dado se calcula mediante la división A/B, donde A es el número total de citas a los artículos publicados por la revista/conferencia en los dos años anteriores y B es el total de artículos publicados por la revista/conferencia en esos dos años. Actualmente CiteSeer^X indexa más de un millón de artículos y registra más de 22 millones de citas a los artículos indexados.

En la Figura 2 mostramos los factores de impacto calculados por CiteSeerX en el gráfico hemos agrupado en años, revistas y conferencias. Para esto, bajamos las páginas Web organizadas por año de la sección (\Venue Impact Ratings" de citeseerx.ist.psu.edu/stats/venues) y a estas páginas les aplicamos scripts para contabilizar los factores de impacto asignados a conferencias y revistas por CiteSeerX. En los archivos HTML se puede detectar sin error cuando se trata de una revista o conferencia. Los resultados muestran que las conferencias tienen factores de impacto comparables a los de las revistas cuando no se hace diferencia entre áreas específicas.

En la Tabla 1 mostramos un ranking según valor de factor de impacto de revistas y conferencias conocidas en distintas áreas de ciencia de la computación. Cada columna representa el ranking para los años 2000, 2002, 2004 y 2006. El valor \1" indica que

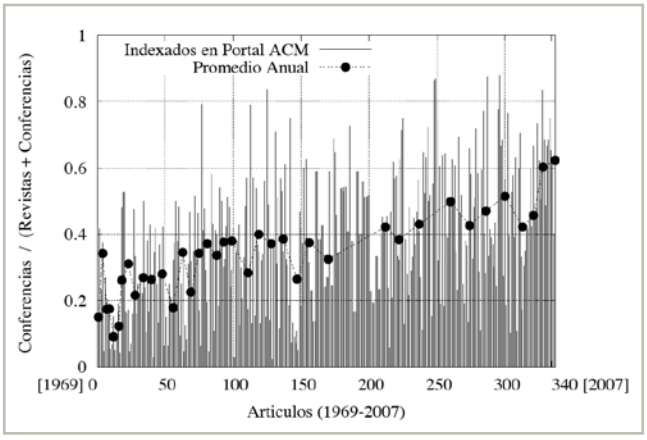


Fig. 1 Proporción de artículos de conferencias citados en cada artículo del ACM Computing Surveys, los cuales están indexados por el portal ACM Digital Library.

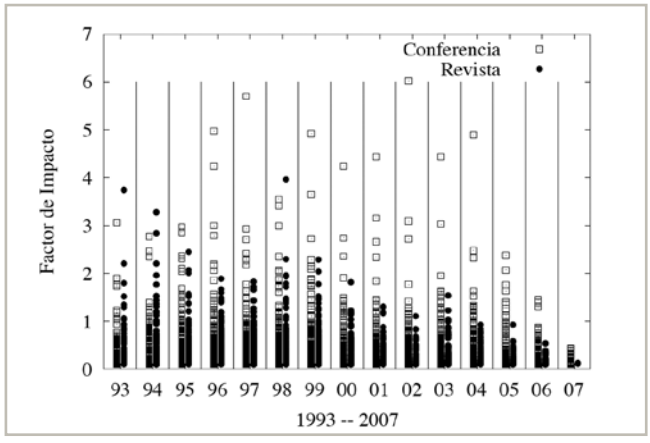


Fig. 2 Factores de impacto tipo ISI para las conferencias y revistas indexadas por CiteSeerX.

Medio	00	02	04	06
Revistas				
TPDS	1	3	2	2
JPDC	3	4	5	5
P.Comp	4	6	6	6
Conferencias				
SPAA	1	3	2	2
IPDPS	3	4	5	5
Euro-Par	4	6	6	6

(a)

Medio	00	02	04	06
Revistas				
TIS	5	2	3	4
TDBS	-	3	4	6
VLDB J.	4	5	6	5
Conferencias				
VLDB	3	1	2	1
SIGIR	1	4	5	3
PODS	2	6	1	2

(b)

Medio	00	02	04	06
Revistas				
JMLR	-	-	3	1
CI	5	5	6	6
ML	2	4	5	5
Conferencias				
IJCAI	4	1	4	4
ICML	1	3	1	2
ACL	3	2	2	3

(c)

Medio	00	02	04	06
Revistas				
TOPLAS	4	4	4	3
IEEE TSE	5	5	5	5
Sci.C.Prog.	6	6	6	6
Conferencias				
PLDI	1	1	3	1
POPL	2	3	1	2
OOPSLA	3	2	2	4

(d)

Tabla 1 Ranking según factor de impacto ISI entre revistas y conferencias en distintas áreas de ciencia de la computación para los años 2000, 2002, 2004 y 2006. (a) Computación Paralela y Distribuida, (b) Bases de Datos y Recuperación de la Información, (c) Inteligencia Artificial (\Machine Learning") y (d) Lenguajes de Programación.

la respectiva revista o conferencia tiene el mayor factor de impacto en su grupo de tres revistas y tres conferencias de la misma área y el valor \6" indica el menor valor de este factor.

VALIDACIÓN DE "CITESEER^{X}" UTILIZANDO "DBLP"

El sistema DBLP Computer Science Bibliography (http://www.informatik.uni-trier.de/ ley/db) mantiene una colección que actualmente contiene sobre un millón de referencias bibliográficas. La base de datos es actualizada con una frecuencia que está prácticamente dentro de la semana en que se publican los nuevos números de las revistas y proceedings de conferencias. En particular es posible bajar desde DBLP un archivo XML que contiene en un formato bien definido los detalles de cada artículo indexado en especial el título, el lugar de publicación escrito de manera consistente para toda la colección y si se trata de un artículo de conferencia o revista.

Para validar los resultados de la Figura 2 bajamos desde CiteSeerX otra sección de este sistema, la cual presenta los diez mil artículos más citados (http://citeseerx.ist.psu.edu/stats/articles). En estos archivos HTML es posible detectar sin error la cantidad de citas que ha recibido el artículo y su título, pero no así el lugar de publicación. Utilizando el archivo XML bajado desde DBLP y los títulos es posible determinar los lugares donde fueron publicados los artículos de CiteSeerX.

Acumulando el total de citas de los diez mil artículos más citados en las respectivas revistas y congresos donde fueron publicados, podemos establecer un ranking de los medios de publicación que consitan el mayor número de citas según CiteSeerX. En la Figura 3 se muestran los resultados, los cuales provienen desde poco más de 200 revistas distintas y casi 400 conferencias. Dichos resultados muestran que un gran porcentaje de las conferencias superan en cantidad acumulada de citas a las revistas. Los artículos de conferencias pueden recibir un número relevante de citas al igual que los artículos de revistas, lo cual se ve reflejado en los factores de impacto ISI mostrados en la Figura 2. De hecho, dentro de los 10 mil artículos más citados obtuvimos el valor 1,04 para la división entre el total de artículos de conferencias y el total de artículos de revistas, y 1,48 al dividir la suma de la cantidad de citas que reciben las conferencias (numerador) y revistas (denominador).

Además el archivo XML de DBLP permite determinar el total de artículos publicados por cada revista o conferencia donde fueron publicados los diez mil artículos más citados en CiteSeerX. Con esto se puede determinar el valor C/P donde C es el total de citas recibidas por los artículos publicados por una conferencia/revista y P es el total de artículos publicados por dicho medio. Es decir, el valor C/P es el promedio de citas que reciben los artículos publicados por

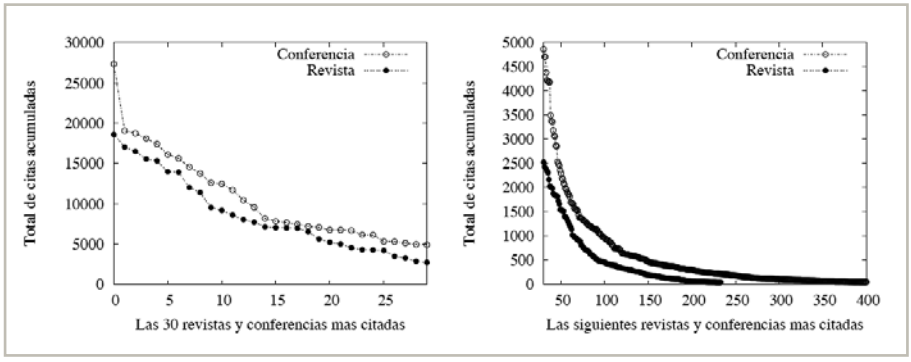


Fig. 3 Total acumulado de citas por conferencia y revista utilizando CiteSeer y DBLP.

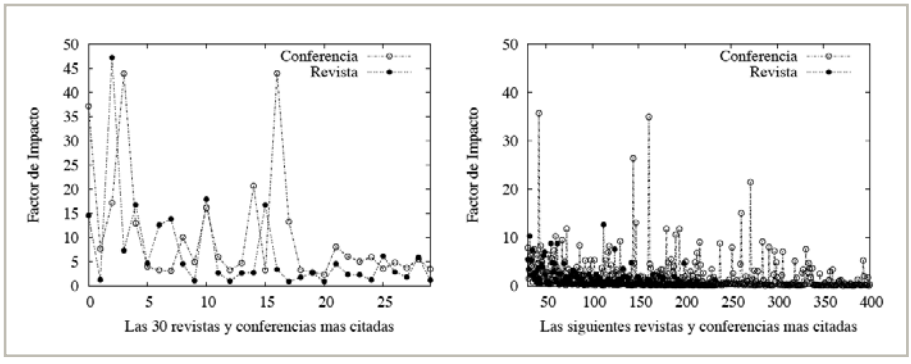


Fig. 4 Número promedio de citas por artículo de conferencia/revista en CiteSeer y DBLP.

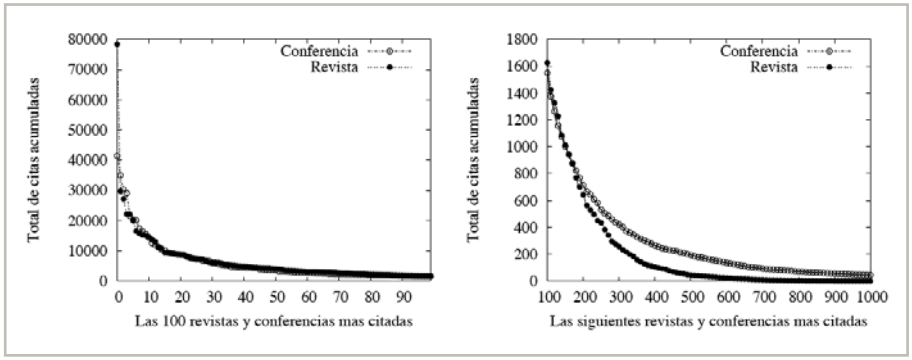


Fig. 5 Total acumulado de citas por conferencia y revista utilizando los artículos indexados en el Portal ACM Digital Library.

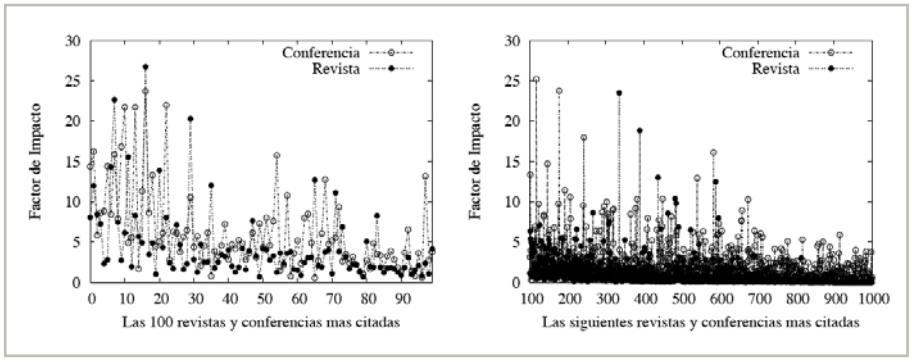


Fig. 6 Citas promedio por conferencia/revista según en el Portal ACM Digital Library.

la revista/conferencia y es similar al índice de impacto ISI, pero considerando todos los años en que el respectivo medio ha publicado artículos. La Figura 4 muestra el factor de impacto C/P que los diez mil artículos más citados le otorgan a la revista o conferencia donde fueron publicados. El orden en que aparecen los datos en el eje X de la Figura 4 es el mismo que el orden dado en la Figura 3.

VALIDACIÓN DESDE EL PORTAL "THE ACM DIGITAL LIBRARY"

El Portal de la ACM Digital Library (portal.acm.org) es un sistema especializado en literatura técnica para ciencia de la computación que contiene una colección de sobre el millón de artículos con sus respectivas listas de referencias bibliográficas e información sobre los artículos que citan a cada artículo. En particular por cada artículo se indica el número total de citas que ha recibido. También es posible diferenciar entre artículos de conferencias y revistas, y la notación de los nombres de cada conferencia/revista es consistente a través de todos los artículos. Utilizamos un cluster de cien procesadores, donde cada uno ejecutó el comando "wget" sobre un URL distinto del Portal ACM para bajar las páginas HTML que son el resultado de ejecutar una búsqueda "vacía" en el Portal. Cada página da acceso a los títulos, autores, "venue" (revista/conferencia) y "citation count" de los 1.233.937 artículos indexados por el Portal al 30 de diciembre de 2008.

Sobre los 62 mil HTML bajados desde el Portal ejecutamos también en paralelo 100 scripts idénticos para obtener por cada revista/conferencia el total de artículos publicados y el total acumulado de citas que recibe cada revista/conferencia a través de sus artículos. Los scripts detectaron un total de 439 mil artículos de conferencia y 456 mil artículos de revistas con más de una cita. El total de citas a los artículos de conferencia es de 912 mil mientras que el total de citas a los artículos de revistas es de 854 mil. Nuestros scripts también detectaron un total de 2.428 conferencias y 1.138 revistas distintas. Los resultados para el total de citas acumuladas por cada revista/conferencia y el número promedio de citas que recibe cada artículo de cada revista/conferencia (*C/P*), para las primeras mil revistas/conferencias se muestran en las Figuras 5 y 6 respectivamente. La tendencia es similar a los resultados obtenidos con las otras muestras de artículos en ciencia de la computación. Es decir, los artículos de conferencia pueden tener una relevancia similar en el avance del estado del arte de la disciplina que los artículos de revistas.

UN ÚLTIMO DATO DESDE "DBLP"

En el XML que bajamos desde DBLP (http://dblp.uni-trier.de/xml) en septiembre de 2008 encontramos que para 28 libros, 6.406 artículos de conferencia y 1.813 artículos de revista, todos publicados antes del año 2005, se incluyen sus respectivas listas de referencias bibliográficas correctamente formateadas con tags XML (actualmente el DBLP no almacena en su base de datos la lista de referencias de los artículos que indexa). Los artículos para los cuales encontramos sus listas de referencias provienen de 22 conferencias y ocho revistas que pertenecen principalmente al área de Bases de Datos. Esto representa una gran oportunidad para analizar lo que sucede en un área clásica y de las más antiguas e importantes de ciencia de la computación. Los libros son

ediciones que están entre los años 1983 y 2004, y las citas en sus listas de referencias abarcan artículos/libros publicados entre los años 1949 y 2001. Los artículos provienen de conferencias anuales que van desde los años 1975 al 2001 y citan artículos/libros publicados desde 1962. Las revistas son números que van desde 1970 al 2001 y citan artículos/libros desde 1945. Las referencias bibliográficas citan artículos publicados en poco más de 150 revistas y 300 conferencias.

En la Figura 7 se muestra el total acumulado de citas por medio de publicación que provienen de los artículos publicados en las conferencias y revistas mencionadas en las listas de referencias bibliográficas de los 28 libros, y todas las ediciones anuales de 22 conferencias y ocho revistas. En la Figura 8 se muestra el promedio de citas por artículo (C=P) que reciben las conferencias y revistas a través de esos artículos. En general, estos resultados muestran la misma tendencia observada en los resultados presentados en las secciones anteriores.

COMENTARIOS FINALES

Tal vez es verdad que gran parte de los avances en ciencia de la computación provienen efectivamente desde artículos presentados en las conferencias más exigentes de cada área de la disciplina. En otras ciencias este tipo de publicaciones no tienen mayor impacto, incluso entre los mismos investigadores de ciencia de la computación no existe consenso al respecto. Sin embargo, los resultados presentados en este artículo indican que para al menos nuestra disciplina este tipo de publicaciones sí tiene importancia. Validamos los resultados utilizando distintas muestras de la literatura indexada por sistemas ampliamente conocidos por la comunidad. En particular, estos sistemas indexan revistas y conferencias que a nuestro juicio están claramente ubicadas en lo que constituye el core de la disciplina tal como se entiende en la iniciativa impulsada en http://www.core.edu.au/.

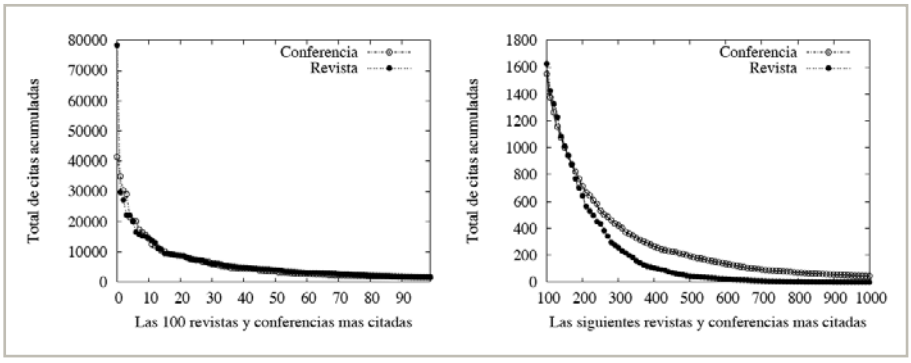


Fig. 7 Total de citas acumuladas por artículos de cada conferencia y revista.

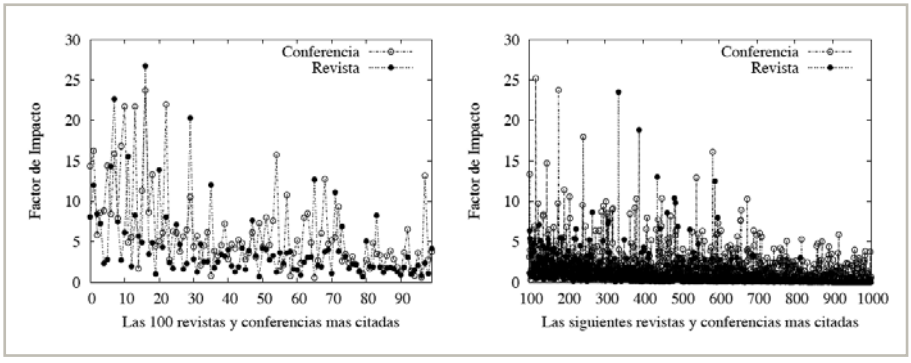


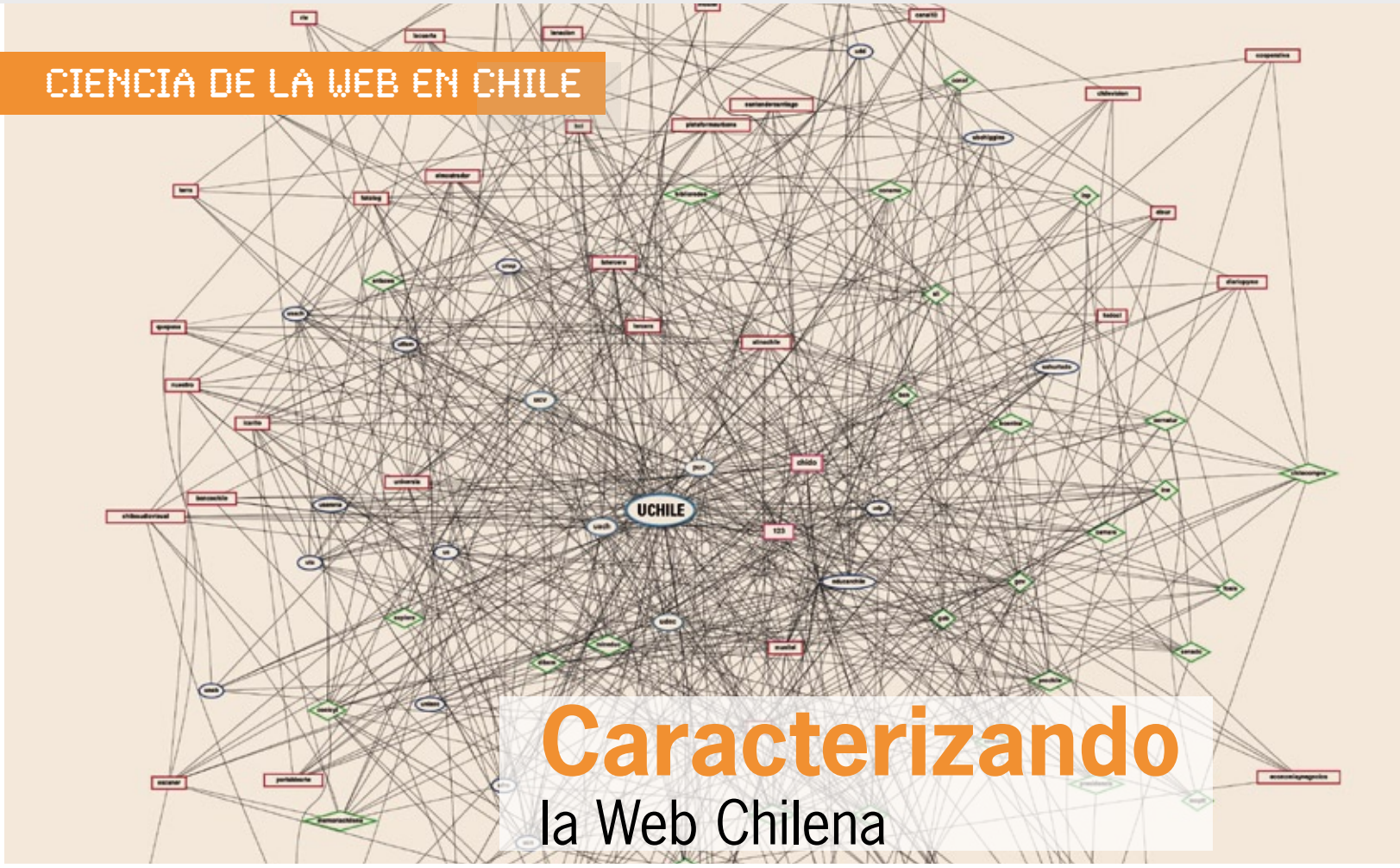
Fig. 8 Factores de impacto de cada conferencia y revista.

AGRADECIMIENTOS

Las revistas y conferencias mencionadas en la Tabla 1 fueron seleccionadas por John Atkinson de la Universidad de Concepción, y Pablo Barceló y Éric Tanter de la Universidad de Chile. Andrea Rodríguez, de la Universidad de Concepción, colaboró con varias sugerencias en distintos puntos de este artículo y ha aportado nuevas estadísticas y vistas de la información para una versión más completa del presente artículo. Senen González, estudiante de Doctorado del DCC, colaboró con los scripts y organización de las máquinas utilizadas para bajar el Portal de la ACM y el procesamiento de los datos. Silvia Menichetti, de la Universidad de Magallanes, colaboró con los scripts utilizados en el ACM Computing Surveys.BITS

Este trabajo ha sido parcialmente financiado por la Sociedad Chilena de Ciencia de la Computación.

CIENCIA DE LA WEB EN CHILE



Caracterizando la Web Chilena

Eduardo Graells
Estudiante de Magíster en Ciencias mención Computación, DCC, Universidad de Chile. Ingeniero Civil en Computación de la misma Universidad.
egraells@dcc.uchile.cl

Ricardo Baeza-Yates
Profesor Titular, DCC, Universidad de Chile. Profesor ICREA Asociado de la Universitat Pompeu Fabra, Barcelona. Ph.D. en Computer Science, University of Waterloo. Vicepresidente de Yahoo! Research para Europa, Medio Oriente y Latinoamérica.
rbaeza@dcc.uchile.cl

INTRODUCCIÓN

La Web es más que un simple conjunto de documentos en distintos servidores, ya que existen relaciones de información entre los documentos mediante los enlaces que se establecen entre ellos. Esto mejora la experiencia de navegación de los usuarios porque los ayuda a encontrar información, y ayuda a los programas que recorren la Web que buscan enlaces de nuevos documentos dentro del contenido de los documentos ya revisados. De ese modo funcionan los motores de búsqueda: permiten encontrar documentos que contengan ciertas palabras claves, y dichos documentos se encontraron siguiendo enlaces. Los resultados de la búsqueda se presentan ordenados de acuerdo a la cantidad de enlaces que reciben (entre otros parámetros que afectan el ordenamiento). Esto permite interpretar el número de enlaces que recibe un documento como una medida de su calidad, porque en general una página enlaza a otras similares.

La Web Global se puede considerar como un gran grafo que tiene una estructura que se puede clasificar como *red libre de escala*, que, al contrario de las *redes aleatorias*, se caracteriza por una distribución dispareja de enlaces, en la que los nodos altamente enlazados actúan como centros que conectan muchos de los otros nodos a la red. Analíticamente, este comportamiento disparejo se puede expresar mediante una ley de potencias (*powerlaw*):

$$frecuencia \approx kx^{-\theta}$$

donde k es una constante que depende del contexto, x es el número de enlaces y $-\theta$ es el parámetro de la distribución. Esto quiere decir que la distribución de

los enlaces es muy sesgada: unas pocas páginas reciben muchos enlaces mientras que la mayoría recibe muy pocos o incluso ninguno. En este artículo se muestra que dicha distribución se puede aplicar a una gran cantidad de aspectos de la Web Chilena, validando la consideración de la Web Global como similar a una red libre de escala, porque estas son auto-similares: una pequeña muestra mantiene características de la red completa.

La Web Chilena se define como el conjunto de sitios cuyo dominio de primer nivel es .cl, o que estén hospedadas en un servidor cuya dirección IP está asociada a Chile. Entre los años 2000 y 2007 sus características han sido estudiadas por el Centro de Investigación de la Web (CIW) y el buscador TodoCL[1]. Esta caracterización se realiza mediante una recolección o *snapshot* de la Web Chilena en un momento particular en el tiempo. La ultima recolección se realizó en septiembre del año 2007, y luego en octubre del mismo año, se realizó una recolección considerando solamente los sitios que tenían enlaces entrantes o salientes, con el fin de obtener una mejor caracterización de los dominios y sitios con más enlaces.

A pesar de estar estudiando un subconjunto acotado de la Web Global, las propiedades que se han encontrado en la Web Chilena son similares a las globales en términos de las distribuciones mencionadas.

A continuación describimos la colección y luego presentamos los resultados a nivel de páginas, sitios y dominios, terminando con las conclusiones.

COLECCIÓN DE LA WEB CHILENA 2007

Habiendo definido qué es la Web Chilena, es necesario recorrerla para obtener los documentos y sitios que la componen. Para realizar la colecta se utilizó el *crawler* WIRE 0.14[2], que a partir de una lista inicial de sitios (semillas o *seeds*) comienza a descargar sus documentos, con el fin de almacenarlos, analizar su contenido y encontrar enlaces a más sitios que son agregados a la lista de sitios por descargar. El proceso se repite hasta que se han descargado todos los documentos públicos posibles, cuando se han agotado algunos parámetros (como el espacio en disco o el límite de documentos a descargar), o cuando se ha llegado a un punto en el cual sólo se están descargando documentos redundantes (situación que ocurre debido a la generación de páginas y URLs dinámicas).

Para la colección que se utilizó en la caracterización del año 2007, se utilizó un computador con una CPU Intel Pentium IV de 3 GHz, 1 GB de memoria RAM y sistema operativo Ubuntu 7.04. El Cuadro 1 resume las características principales de la colección.

Páginas Web	9.637.801
Texto en total	135,76 [GB]
Texto promedio por página	14,77 [KB]
Sitios Conocidos	200.000
Sitios Recolectados	111.374
Páginas promedio por sitio	86,53
Texto promedio por sitio	1,24 [MB]
Dominios Conocidos	190.577
Dominios Recolectados	104.409
Sitios promedio por dominio	1,07
Páginas promedio por dominio	92,31
Texto promedio por dominio	1,33 [MB]

Cuadro 1 Resumen de estadísticas de la colecta.

CARACTERÍSTICAS DE LOS DOCUMENTOS

La caracterización de la Web Chilena contempla 9.637.801 documentos o páginas Web que fueron descargadas desde la Web pública y/o visible. Una página promedio pesa 14,77 KB sin considerar imágenes u otros contenidos multimedia incrustados en ella, aunque los enlaces a archivos multimedia o documentos de texto son registrados para su posterior análisis. En el tamaño de las páginas es donde se encuentra la primera ley de potencias: la distribución de tamaño de los documentos versus la fracción de los documentos sigue una ley de potencias con parámetro -3,56 para páginas de más de 40 KB, y de -0,82 para páginas entre 11 y 40 KB.

Al descargar cada página, el servidor Web da a conocer la fecha en la cual dicha página se modificó por última vez. Este dato nos permite modelar la distribución de la edad de los documentos, en la cual nos damos cuenta que en los 12 meses previos a la recolección se actualizó un 19% de los documentos de la colección. Naturalmente también hay documentos que no se actualizan hace mucho tiempo, pero esa cantidad está acotada por los documentos encontrados en las colectas anteriores. Esta descripción es propia de una ley de potencias: la distribución de la edad de los documentos se puede aproximar con una ley de parámetro -1,27.

> La cantidad de servidores que no entregan información no es despreciable, aunque en general se puede concluir que en los servidores las tecnologías de código abierto superan ampliamente a las tecnologías propietarias.

[1] http://www.todocl.cl
[2] http://www.cwr.cl/projects/WIRE

De las páginas descargadas, 34% de ellas son dinámicas; páginas generadas en el momento de ser solicitadas sin que existieran previamente. Esto es muy común en la actualidad; el contenido de un sitio se almacena en una base de datos y las páginas en realidad son programas que leen dicho contenido al momento de ser solicitada la página por el usuario. Dichos programas consideran distintas variables, como aquellas entregadas en la URL o una posible autenticación previa del usuario en el sitio. La aplicación más usada para generar estas páginas es PHP[3], una tecnología de código abierto, que tiene un 79,36% de participación. Le sigue la tecnología ASP, propietaria y de plataforma restringida, con un 18,07%.

Respecto a enlaces a documentos no HTML, estos se pueden agrupar por tipo:

- **Texto:** 1,5 millones de enlaces. Los formatos más populares son PDF (56,74%), XML (26,69%) y DOC (6,51%).
- **Imagen:** 100 millones de enlaces. Los formatos más populares son GIF (77,26%), JPG (18,26%) y PNG (4,45%)
- **Audio:** 166 mil enlaces. Los formatos más populares son WMA (40,29%) y MP3 (39,23%).
- **Vídeo:** 35 mil enlaces. Los formatos más populares son WMV (49,59%), QT (18,20%), MPEG (10,65%) y RM (10,54%).

CARACTERÍSTICAS DE LOS SITIOS

La recolección se inició con una lista de seeds o semillas que contenía todos los dominios .cl registrados, así como sitios no .cl que se conocían de colectas anteriores. La lista de dominios la provee NIC Chile gracias a un acuerdo de investigación. A cada uno de esos dominios se les agrega el prefijo www, ya que en general es correcto asumir que las direcciones http://sitio.cl y http://www.sitio.cl apuntan al mismo sitio.

En el proceso de recolección se llegó a conocer un total de 200.000 direcciones de sitios, aunque solamente 111.374 pudieron ser recolectados. Aquellos que no pudieron ser recolectados no tenían una dirección IP asociada al momento de realizar la recolección. Así, un sitio tiene en promedio 86,53 páginas y un contenido HTML total de 1,24 MB. Sin embargo, estas distribuciones también presentan leyes de potencia, lo que indica que existen muchos sitios con pocas páginas o contenido y pocos sitios que agrupan una gran cantidad de páginas y del tamaño total de la Web Chilena: sólo un 7% de los sitios tiene el 90% de los documentos, y un 14% de los sitios contiene el 99% del total del contenido. Los parámetros de dichas leyes de potencia son -1,84 para el número de páginas por sitio y -1,64 para el tamaño de los sitios.

Los enlaces de entrada y salida que tiene un sitio también son estudiados. Se definen como grado interno y grado externo: el grado interno de un sitio S es el número de sitios distintos que enlaza a S, y el grado externo de S es el número de sitios distintos enlazados por S. Es decir, dentro del grado interno es indiferente si distintas páginas dentro del mismo sitio son enlazadas por otro o enlazan a otro, en ambos casos, todos esos enlaces sólo incrementan en uno el grado correspondiente. Este esquema de enlaces forma un grafo de sitios de nido como *Hostgraph*. En el Hostgraph la distribución del grado interno y externo para los sitios es muy sesgada, ya que hay pocos sitios que reciben una gran cantidad de enlaces y muchos que reciben pocos o incluso ninguno. Las leyes de potencias que modelan estas distribuciones tienen parámetros -2,16 (grado interno) y -2,32 (grado externo)[4].

El Cuadro 2 muestra los 5 sitios más destacados en el número de páginas tamaño en GBs, grado interno y grado externo. Los sitios con mayor grado interno han mantenido su posición constantemente a lo largo de los años, mientras que los sitios con más páginas y contenido cambian cada año.

PÁGINAS	CONTENIDO (GB)	GRADO INTERNO	GRADO EXTERNO
www.autovia.cl (22.825)	www.suena.cl (1,67)	www.sii.cl (542)	www.chido.cl (1.253)
www.b2.cl (22.473)	www.amazon.cl (1,55)	www.uchile.cl (398)	www.fotolog.cl (523)
www.ais.cl (22.100)	www.planetashile.cl (1,15)	www.mineduc.cl (374)	www.atinachile.cl (416)
www.kontent.cl (21.613)	listados.deremate.cl (0,91)	www.meteochile.cl (335)	www.todocl.cl (352)
www.madness.cl (21.244)	www.b2.cl (0,85)	www.corfo.cl (290)	www.webs.cl (292)

Cuadro 2 Sitios destacados en las distintas variables analizadas.

ESTRUCTURA DE LA WEB CHILENA

En un grafo, una componente fuertemente conectada (SCC por *Strongly Connected Component*) es aquella en la que se puede llegar desde un nodo hasta cualquier otro siguiendo las aristas en el grafo, respetando la dirección de éstas. Este análisis se puede aplicar al hostgraph, y es así como se encuentra una SCC gigante y una cantidad menor de componentes fuertemente conectadas más pequeñas (incluyendo SCCs de tamaño 1). El tener una SCC gigante es un signo típico de redes libres de escala.

La SCC gigante de la Web Chilena tiene 6275 sitios, y puede considerarse como el punto de partida desde el cual se define la estructura de la Web Chilena.

[3] http://www.php.net

[4] Estos valores corresponden a la colecta de septiembre. Para la realizada en octubre, los valores son -1,83 y -1,84 para los grados interno y externo, respectivamente.

En base a la conectividad que tienen los sitios con aquellos presentes en la SCC gigante, se pueden definir las siguientes componentes:

- **MAIN**, los sitios en la componente fuertemente conexa.
- **OUT**, los sitios que son alcanzables desde MAIN, pero que no tienen enlaces hacia MAIN.
- **IN**, los sitios que pueden alcanzar a MAIN, pero que no tienen enlaces desde MAIN.
- **ISLAS**, sitios desconectados de los demás en términos de enlaces.
- **TENTÁCULOS**, sitios que sólo se conectan con IN o OUT, pero en el sentido inverso de los enlaces.
- **TÚNEL**, una componente que une las componentes IN y OUT sin pasar por MAIN.

La componente MAIN se extiende en las siguientes subcomponentes:

- **MAIN-MAIN** son los sitios que pueden ser alcanzados directamente desde la componente IN o que pueden alcanzar directamente la componente OUT.
- **MAIN-IN** son los sitios que pueden ser alcanzados directamente desde IN pero no están en MAIN-MAIN.
- **MAIN-OUT** son los sitios que pueden alcanzar directamente a OUT pero no pertenecen a MAIN-MAIN.
- **MAIN-NORM** son los sitios que no pertenecen a las subcomponentes definidas anteriormente.

La Figura 1 muestra una representación gráfica de la estructura descrita y la distribución de sitios y páginas a través de las componentes. La componente ISLAS contiene la mayor cantidad de sitios, pero la que tiene más páginas es MAIN (en particular MAIN-MAIN). Esta estructura permite encontrar relaciones entre los sitios que pertenecen a cada componente: usualmente en IN se encuentran portales o sitios de inicio en la red, que buscan enlazar a sitios importantes; en OUT se encuentran sitios que buscan recibir enlaces pero que no entregan enlaces a otros sitios; en MAIN suelen haber sitios interconectados entre sí como pueden ser sitios de universidades y del gobierno. En la componente ISLAS se encuentra la mayoría de los sitios con una única página recolectada, aunque es posible que si se consideran las páginas que no se recolectaron en esos sitios se encuentren enlaces que permitan llevar a algunos de los sitios a IN o a TENTÁCULOS. Adicionalmente, en MAIN se encuentra la mayoría de los enlaces a documentos de texto en formato no HTML.

CARACTERÍSTICAS DE LOS DOMINIOS

Si bien se conocen 190.577 dominios distintos, solamente se pudieron recolectar 104.409, aunque de éstos se pudieron contactar 117.700. Esto quiere decir que hay dominios que tienen una dirección IP asociada pero no tienen ninguna página (o bien tienen páginas, pero éstas son privadas), ya que al tratar de conectarlos entregan códigos de error (como pueden ser *404 - Not Found* o *403 - Forbidden*). En total se encontraron 14.477 direcciones IP. La distribución de los dominios en estas direcciones también sigue una ley de potencias, de parámetros -0; 35 en su parte inicial y -1; 37 en su parte central. La distribución es tan sesgada que 2 direcciones tienen más de 1.000 dominios, y 9.026 direcciones tienen solamente 1 dominio cada una.

Para cada dirección IP se le pidió al servidor información sobre el software instalado mediante un *Request HTTP*. Como resultado se obtuvo lo siguiente:

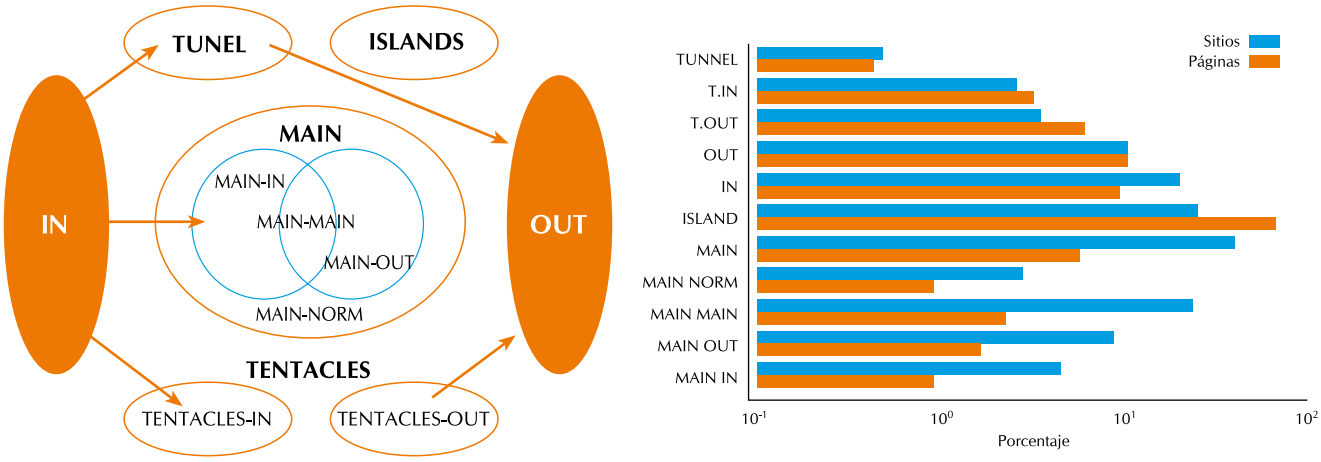


Fig. 1 A la izquierda, una representación gráfica de la estructura de la Web Chilena. A la derecha, la distribución de sitios y páginas en las componentes (en escala logarítmica).

- **Sistema Operativo:** 43,21% no entrega información, 38,67% usa GNU/Linux y Unix, 18,12% usa Microsoft Windows.
- **Servidor Web:** 38% no entrega información, 43% usa Apache, 18,12% usa Microsoft IIS.

La cantidad de servidores que no entregan información no es despreciable, aunque en general se puede concluir que en los servidores las tecnologías de código abierto superan ampliamente a las tecnologías propietarias. Esto es coherente con los resultados obtenidos en la identificación de tecnologías para páginas dinámicas.

Respecto a las características analizadas en las secciones anteriores, el Cuadro 3 muestra los 5 dominios más destacados en cantidad de sitios, contenido y grado interno. En general, el comportamiento de los dominios es más estable que el de los sitios, aunque también se pueden encontrar anomalías, como puede verse en los dominios más enlazados, donde existen tres dominios que apuntan al mismo sitio, correspondiente a una protección de dominios.

En base a los enlaces entre dominios se ha creado una representación gráfica de la Web Chilena, visible en la Figura 2. Para esta representación se eligieron los 31 dominios más conectados entre sí, considerando de la lista de dominios más enlazados solamente aquellos que tenían sitios en la componente MAIN MAIN. Los dominios son representados como nodos enlazados por una línea cuyo grosor y color muestra la cantidad de enlaces entre ellos (mientras más oscuro y grueso, hay una mayor cantidad de enlaces). Se dividen en tres grupos: comerciales (rectángulos), de instituciones educacionales (elipses) y de gobierno (rombos). En la imagen se aprecia que dominios del mismo tipo tienden a estar más cercanos entre sí.[5]

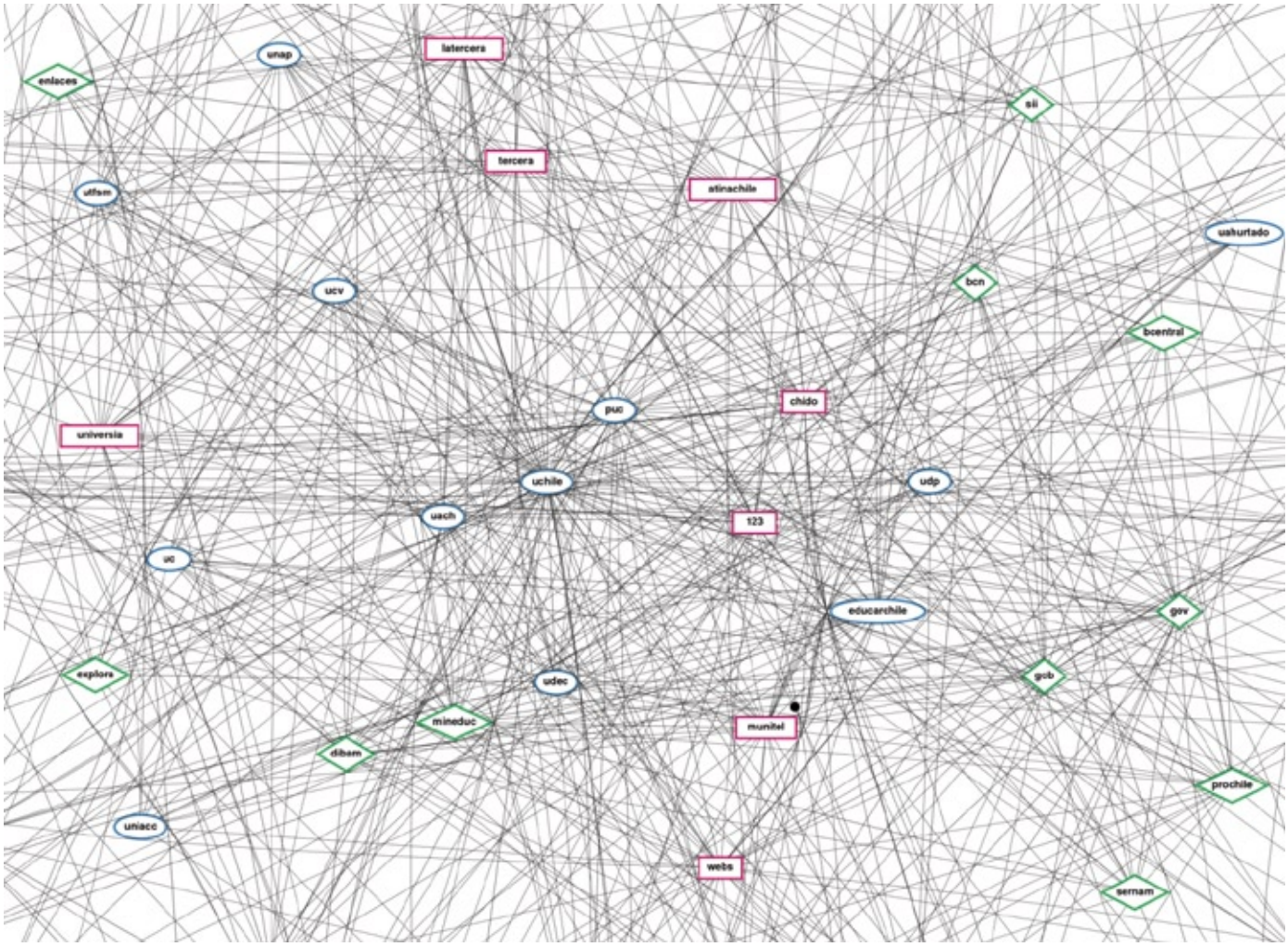


Fig. 2 Una visión gráfica de los 31 dominios más conectados entre sí de la Web Chilena.

[5] En el reporte técnico "Características de la Web Chilena 2007" se observa esta imagen cabalmente, con más de 100 dominios.

CONCLUSIONES

La Web Chilena ha cambiado bastante respecto a los últimos años y, a pesar de estar en constante cambio, sigue manteniendo una estructura similar a la encontrada en años anteriores. El crecimiento en la cantidad de documentos recolectados desde el año 2006 es notorio, desde 7; 4 millones a 9; 6 millones, lo cual es consecuente con la cantidad de documentos creada o actualizada en los últimos 12 meses. La distribución de los documentos en diferentes análisis se puede ajustar a leyes de potencias, verificando el modelo de redes libres de escala.

Aunque se conocía la dirección de 200;000 sitios, sólo se pudieron recolectar cerca de 111;000. El análisis de algunas características de los sitios también presenta leyes de potencias: la distribución de documentos por sitios, la del contenido y la de enlaces entre sitios. Además, los sitios que reciben más enlaces se han mantenido a lo largo de los años, y destacan por ser sitios del gobierno, de instituciones educacionales o de medios de comunicación. La macroestructura de la web también presenta características importantes: aunque solamente un 5% de los sitios recolectados está fuertemente conectado entre sí, estos sitios tienen el 39% del total de las páginas. A su vez, un 65; 26% de los sitios está aislado de los demás, y contienen cerca del 24% del total de las páginas.

SITIOS	CONTENIDO (GB)	GRADO INTERNO
portalciudadano (690)	turismo-viajes (3,04)	uchile (1.300)
uchile (374)	suena (1,68)	nameaction, backorder, snapnames (906, 904, 902)
scd (352)	deremate (1,63)	gov (653)
loquegustes (342)	amazon (1,55)	puc (550)
boonic (267)	mercadolibre (1,55)	sii (542)

Cuadro 3 Dominios destacados en las distintas variables analizadas. Los dominios nameaction, backorder y snapnames son *mirrors*.

La distribución de direcciones IP para los dominios también se ajusta a una ley de potencias. En estas direcciones se estudió la tecnología que utilizaba el servidor y, en las que entregaron información, se encontró que tanto en el sistema operativo como en el servidor utilizado las tecnologías de código abierto tienen mayor presencia.

Se concluye que la caracterización de una Web Nacional, en este caso la Web Chilena, permiten, además de establecer un modelamiento de la Web en términos matemáticos o analíticos, tener datos concretos que sirven de base para estudios de usabilidad, de mercado y de minería de datos, entre otros. Lo que se ha realizado es una captura de un instante particular de la existencia de la Web, cuya representatividad no se puede poner en duda al ver la constancia que se ha tenido durante los pasados años y los resultados similares vistos en estudios aplicados a otras Web Nacionales.BITS

REFERENCIAS

- Centro de Investigación de la Web - `http://www.cwr.cl` Ver Estudios de la Web Chilena.
- TodoCL - `http://www.todocl.cl`
- Albert-Lászl o Barabási. *Linked: The New Science of Networks*. Perseus Books Group, Mayo 2002.
- A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, y J. Wiener. Graph structure in the web: experiments and models. *Proceedings of the ninth WWW Conference*, 2000.
- Ricardo Baeza-Yates y Bárbara Poblete. Dynamics of the chilean web structure. *Computer Net-works*, Julio 2006.
- Eduardo Graells y Ricardo Baeza-Yates. Evolution of the chilean web: a larger study. *Proceedings of the sixth LA-WEB Conference*, Octubre 2008.
- Ricardo Baeza-Yates, Carlos Castillo, y Efthimis Efthimiadis. Characterization of national web domains. *ACM TOIT*, 2006.

CIENCIA DE LA WEB EN CHILE

Panorama de la Investigación sobre la Web en Chile

Mauricio Marín
Sociedad Chilena de Ciencia de la Computación. Yahoo! Research Latin America, Universidad de Chile. PhD en Computer Science, University of Oxford,UK.
mmarin@yahoo-inc.com

Claudio Gutiérrez
Profesor Asociado, DCC, Unviersidad de Chile. Ph.D. en Computer Science de Wesleyan University, Estados Unidos. Investigador Asociado del Centro de Investigación de la Web y el Grupo Khipu de bases de datos.
cgutierr@dcc.uchile.cl

PREHISTORIA

La investigación relacionada a las Tecnologías de la Información (TI) en Chile es bastante joven comparada con otras áreas de la ciencia y la ingeniería. Como muestra [2], el número de artículos relacionados a las TI publicados durante los años 80s, por autores en Chile en revistas internacionales con comites editoriales, promediaba los 70 anuales. Esto da un promedio de 4,85 artículos por habitante, el máximo indice por habitante en Latinoamérica (el promedio en Latinoamérica era en ese entonces de 1,59 artículos por persona).

Con respecto a la investigación sobre la web, una buena fotografía del estado del arte a inicios de los '90s es la siguiente frase de un artículo describiendo los mayores logros de las TI en Chile:

"This has allowed the spread of new technologies such as LANs and WANs (ethernets, bitnet, uucp, internet)" [3]

Es decir, las mayores preocupaciones eran la espina dorsal de Internet y la difusión y escalabilidad de las técnicas para redes.

Probablemente el primer artículo técnico sobre los problemas de Internet fue el escrito por R. Baeza-Yates, J.M. Piquer, y P. Poblete, en el cual discutían los problemas asociados a las conexiones a Internet en Chile [4]. Curiosamente, en esta primera investigación aparecieron los mismos tipos de problemas que asombraron a Darwin cuando visitó Chile durante la primera mitad del siglo XIX; es decir, lo complejo de su geografía:

"Chile has had two international 56Kbps links to the Internet since January 1992.

Probablemente el primer artículo técnico sobre los problemas de Internet fue el escrito por R. Baeza-Yates, J.M. Piquer y P. Poblete, en el cual discutían los problemas asociados a las conexiones a Internet en Chile

This online coAAnnection to the world is having a great impact on academic research, extending state-of-the-art communication technology to our country. The impact will be even greater considering the traditional isolation of the country, surrounded by mountains and sea at the end of the world, and a very particular geography: almost 4350 kilometers long from north to south with an average width of 190 kilometers.

En 1995 la palabra web aún no asomaba en el léxico de los investigadores nacionales. En el conocido artículo "Computing in Chile: The Jaguar of the Pacific Rim?", publicado en el Communications of the ACM, la Web no es mencionada. Los problemas aún eran los de Internet y las conexiones, cuya evolución es presentada en los siguientes hitos [5]:

- 1985: Mail electrónico internacional (uucp), seguido por Bitnet en 1987 (U. de Chile).
- 1987: Implementación de la primera red automática de cajeros dispensadores de plata.
- 1991: Inicio de la primera red de datos y de conectividad a Internet (entre la Universidad de Chile y la Universidad Católica).
- 1994: Puesta en funcionamiento de servicio ISDN experimental.

Probablemente el primer artículo en mencionar a la Web fue *A Model for Visualizing Large Answers in WWW* [1], presentado en la Conferencia Chilena de Ciencia de la Computación. El modelo presentado estaba basado principalmente en técnicas de recuperación de información, en las cuales uno de los autores, Ricardo Baeza-Yates, era un reconocido experto internacional.

El siguiente paso fue el estudio de la web chilena, que usó la información de un motor de búsqueda local, *todoCL.cl*, proyecto liderado por el mismo Baeza-Yates. *TodoCL* comenzó a operar en marzo de 2000 en colaboración con otro proyecto similar en Brasil, Akwan. En particular, *TodoCL* ha sido el único motor de búsqueda local cuyo objetivo es la Web chilena, y ha entregado datos para realizar estudios acerca de la caracterización de esta Web y otros estudios basados en *query logs*. Además, esto datos permiten hacer investigaciones sobre la dinámica de la Web que a menudo son impracticables.

EL CENTRO DE INVESTIGACIÓN DE LA WEB (CIW)

El año 2002 marca un punto de quiebre en la investigación de la Web en Chile, con la instalación del *Centro de Investigación de la Web* (CIW). Este centro, dirigido en sus primeros años por Baeza-Yates, nació con el financiamiento del gobierno de Chile a través de Mideplan.

El desafío era doble. Por una parte, desarrollar a nuevos niveles la investigación sobre la Web en Chile y, por otro, desarrollar la necesaria sinergia entre investigadores sin mayor interacción científica previa y dedicados con anterioridad a temas de investigación no directamente relacionados con la Web. Al final el proyecto resultó muy exitoso.

Después de tres años, el CIW fue capaz de reunir a 10 investigadores (2 post-docs), más de 55 estudiantes, y producir cerca de 50 artículos cada año. El campo de investigación de este proyecto era bastante amplio, aunque podía ser agrupado en dos áreas principales: (1) Bases de datos y recuperación de información, y (2) Sistemas distribuidos y redes. A continuación describimos los principales temas de investigación del centro basándonos en un informe interno del CIW del año 2004:

Bases de datos y recuperación de información

Esta área cubre recuperación multimedia, información espacial e información semiestructurada. El tema que subyace a estos tres es el de *combinatorial pattern matching*, un área de investigación que estudia desde un punto de vista combinatorial cómo buscar ciertos patrones en estructuras discretas y regulares como secuencias o grafos. Otro tema es cómo agregar información semántica a los contenidos, como metadatos y la web semántica. Incluimos aquí también el tema de minería de datos en la Web. A continuación describimos cada una de estas subáreas:

- Análisis multimedia y técnicas de búsqueda es un área de investigación en la que había trabajado Ricardo Baeza-Yates, Gonzalo Navarro (DCC, Universidad de Chile), Andrea Rodríguez (Informática y Computación, Universidad de Concepción), y Javier Ruiz del Solar (Ing. Eléctrica, Universidad de Chile). Esta área tenía un postdoc y varios alumnos de Ph.D. y master. En general, cubría todos los problemas de búsqueda relacionados con textos y multimedia, con mayor énfasis en algoritmos espaciales de búsqueda y algoritmos de *string matching*.
- Web semántica fue inicida por Carlos Hurtado (hoy en Universidad Adolfo Ibañez) y Claudio Gutiérrez (DCC, Universidad de Chile). Ellos comenzaron formalizando las especificaciones del Consorcio de la Web y justificando en términos de sus bases de datos. Esta área resultó ser muy exitosa. También relacionado con esto estaba el tema de agregar información semántica a los contenidos Web y obtener información desde ellos utilizando minería de datos.

Crawling y técninas de ranking es un área de investigación en la que Ricardo Baeza-Yates, Mauricio Marín (por entonces en la Universidad de Magallanes) y Andrea Rodríguez trabajaban. El principal tema de investigación era cómo crawlear la Web completa, reuniendo páginas para indexarlas y luego construir motores de búsqueda eficientes. Muchos tradeoffs aparecen cuando se trata de construir uno de estos motores, y el trabajo del grupo se centró en proponer nuevas tecnologías para mejorar la eficiencia y exactitud de la búsqueda.

TodoCL ha sido el único motor de búsqueda local cuyo objetivo es la Web chilena, y ha entregado datos para realizar estudios acerca de la caracterización de esta Web y otros estudios basados en query logs. Además, esto datos permiten hacer investigaciones sobre la dinámica de la Web que a menudo son impracticables.

Sistemas distribuidos y redes

Esta área cubre las tecnologías de programación para aplicaciones Web (*Web Agents, Web Services, Distributed Programming*), protocolos de comunicación para nuevos medios (*Multimedia over IP*), y tecnologías para mejorar el rendimiento de las herramientas de la Web (*parallel search, crawling technologies*).

José Piquer y Éric Tanter (en ese tiempo alumno de doctorado) trabajaron en Agentes móbiles y Programación distribuida, en una plataforma de programación basada en reflexión para Java (Reflex). Mauricio Marín y Gonzalo Navarro trabajaron en paralelismo. Un cluster de 10 nodos fue instalado en el DCC de la Universidad de Chile, sobre el cual se trabajó en el desarrollo de algortimos paralelos para bases de datos de textos, y para el *scheduling* distribuido de servidores simultáneos en la Internet.

Como se puede ver, la investigación no sólo era amplia con respecto a los temas, sino también con respecto a los participantes: El CIW incluyó investigadores de Santiago, Concepción, Punta Arenas, y estudiantes de muchas partes del país.

En el año 2003, por impulso del CIW, y bajo la tutela del *International World Wide Web Conference Committee* (IW3C2) y la Sociedad Chilena de Ciencia de la Computación (SCCC), se organizó en Santiago el Primer Congreso Latinoamericano de la Web. Este evento continuó además durante los siguientes años, convirtiéndose en una de las referencias de los investigadores de la región trabajando en el tema de la Web.

La expresividad y eficiencia son la clave para el éxito de las nuevas aplicaciones Web.

ACTUALIDAD

El CIW fue un éxito total. Fue además instrumental para atraer a Chile, en 2006, al primer laboratorio Yahoo! en el hemisferio sur. También las escuelas de verano realizadas anualmente y las nuevas fuentes de financiamiento atrajeron al CIW una ola de estudiantes a los temas relacionados con la Web. Hoy en día difícilmente hay un departamento de Computación en Chile dedicado a la investigación que no cubra algo ligado a la Web.

Una nueva etapa se abrió en 2006. Gonzalo Navarro reemplazó a Baeza-Yates como director del CIW, quien pasó a estar a cargo del laboratorio de Yahoo! Durante 2004 nuevos post-docs se incorporaron al proyecto: George Dupret and Benjamin Piwowarski cumplieron un importante papel en el área de minera de datos. También nuevos profesores de universidades locales se unieron al centro: Pablo Barceló, Benjamín Bustos y Éric Tanter del DCC de la Universidad de Chile, y Marcelo Arena, del DCC de la Universidad Católica. El centro además incorporó pos-docs desde fuera del país, así como estudiantes de posgrado de fuera de Santiago y de otros países de la región.

Las áreas de investigación se fueron refinando a través de los años. Para dar un panorama de la investigación actual, describiremos las líneas de investigación vigentes del CIW siguiendo su informe interno del año 2008:

- **Estructuras de datos compactas.** El objetivo es tratar de sacar ventaja del creciente gap que existe entre las velocidades de los niveles consecutivos de la jerarquía de memoria, mediante el diseño de estructuras de datos que operen con poco espacio y que por tanto quepan en memorias más rápidas.
- **Recuperación multimedia.** Las consultas que se hacen a las bases de datos multimedia buscan por similaridad más que por exactitud (como se hace en las bases de datos tradicionales). Esta área intenta buscar modelos de similaridad para objetos multimedia que se correspondan con la noción humana e intuitiva de similaridad, y también estructuras de datos y algortimos que apoyen una eficiente búsqueda por similaridad.
- **Lenguajes de programación y ambientes.** Intenta proveer el apoyo adecuado a softwares complejos que realizan *debugging*, a la programacion de ambientes inteligente, y a los aspectos dinámicos de los lenguajes de programación. Todos estos son importantes para el diseño de software para aplicaciones Web.
- **Estudio de la estructura de la Web.** Se intenta entender la dinámica de la Web, por ejemplo su estructura de conectividad, crecimiento y dinámica de cambio, etc. Esto permite un amplio rango de estudios, desde entender muchos fenómenos sociales hasta poder construir aplicaciones Web más eficientes.
- **Lenguajes para bases de datos.** El objetivo es diseñar lenguajes de consulta apropiados para manipular información en la Web, que es más compleja que en las bases de datos tradicionales. Por ejemplo, XML y RDF presentan nuevos desafíos en términos de expresividad y eficiencia, y son la clave para el éxito de las nuevas aplicaciones Web.

- **Consultas complejas para objetos en movimiento.** Las bases de datos de objetos en movimiento son una solución factible al problema de escalabilidad de un sistema centralizado de bases de datos. El problema que se ha tratado en el CIW es el de encontrar métodos de indexamiento distribuido usando un meta índice también distribuido.

El CIW ha ganado en el último tiempo un amplio reconocimiento en la comunidad de invesigación sobre la Web. Indicadores de esto son los 3 premios al mejor artículo obtenidos por investigadores del centro en conferencias relacionadas a la Web, y la apertura del laboratorio de investigación de Yahoo! en Chile.

Yahoo! Research Latin America es un nuevo laboratorio de investigación, localizado en la Escuela de Ingeniería de la Universidad de Chile. Bajo la dirección de Baeza-Yates, este laboratorio se concentra en las áreas de investigación de la Web y minería de datos. A continuación detallamos algunas de las áreas investigadas en el centro:

- **Búsqueda social.** El laboratorio ha estudiado la información obtenida a través de la búsqueda realizada por los usuarios mediante clicks. Básicamente, esta es la información que un usuario entrega al interactuar con su motor de búsqueda, por medio de escribir una consulta, luego cliquear en ciertos documentos, y eventualmente reescribir la consulta original.
- **Motores de búsqueda Sync/Async.** Los motores de búsqueda deben saber lidiar eficientemente con el tráfico de consultas generado por los usuarios. Redundancia de hardware puede ser reducida utilizando estrategias de procesamiento de consultas, en el caso en que un gran número de consultas pueden ser resueltas concurrentemente.
- **Bulk-synchronous crawling.** Los centros de datos de gran escala para los crawlers son capaces de manetener un gran número de conexiones HTTP activas, para poder bajar lo más rápido posible el enorme número de páginas Web desde una sección de la Web especificada. Esto genera un continuo flujo de nuevas URLs de documentos. El laboratorio investiga como se puede manejar eficientemente este problema mediante paralelización.

CONCLUSIONES

La información presentada en este artículo muestra el crucial rol que el CIW ha jugado en crear, desarrollar y consolidar la investigación acerca de la Web en Chile. En este punto es importante remarcar que el CIW no sólo ha sido importante en términos de investigación, sino también en términos de alcance de un público más amplio. En relación a esto podemos mencionar los estudios acerca de la Web chilena, la Ventana Digital que conectó a las ciudades de Arica y Santiago, un concurso para estudiantes acerca de la Web, y finalmente el libro ¿Cómo funciona la Web? que se ha vuelto bastante popular entre los alumnos y profesores de enseñanza media.

Con respecto a la investigación, presentamos un panorama cualitativo del estado del arte de la investigación sobre la Web en Chile. Por cierto, esto no representa toda la investigación sobre la Web realizada en Chile, pues nos hemos concentrado en aquella realizada por el CIW y Yahoo!, que son actualmente las dos mayores fuentes de investigación acerca del tema en el país.

Creemos que un estudio también cuantitativo acerca del tema se hace bastante necesario. Los datos y los investigadores están todos presentes, una oportunidad que no debería ser desperdiciada. BITS

REFERENCIAS

[1] O. Alonso and R. Baeza-Yates, A Model for Visualizing Large Answers in WWW. In *XVIII Int. Conf. of the Chilean CS Society*, 1998.

[2] R. Baeza-Yates, D. Fuller, J. Pino. Innovation as a critical success factor for the development of an information technology industry in Chile. In *12th IFIP World Computer Congress*, 1992.

[3] R. Baeza-Yates, D. Fuller, J. Pino. IT landmarks in less-developed countries: The Chilean case. In *21st CAIS/ACSI Annual Conference*, 1993.

[4] R. Baeza-Yates, J.M. Piquer, P. Poblete. The Chilean internet connection or I never promised you a rose garden, In *INET O93*, 1993.

[5] R. Baeza-Yates, D. Fuller, J. Pino, S. Goodman. Computing in Chile: The jaguar of the Pacific rim?, *Communications of ACM*, 38. 1995.

Entrevista Nacional:
Ricardo Baeza - Yates

Por Gonzalo Navarro

Es, por lejos, el investigador chileno más citado en ciencia de la computación, y el único investigador del área que pertenece a la Academia de Ciencias de Chile. Fue el creador del Centro de Investigación de la Web (CIW), único Núcleo Milenio en la disciplina, y es responsable del establecimiento del primer laboratorio Yahoo! Research en Chile y Barcelona. Hoy es vicepresidente de Investigación para Europa y Latinoamérica de Yahoo! Research, y supervisa los dos laboratorios ya mencionados más uno reciente en Haifa, Israel. Además es un viajero incansable, de actividad frenética en general, sin que por ello deje algo de tiempo para una buena conversación. Quisimos conocer de primera mano la visión de este profesor del DCC sobre la Web, la ciencia de la computación y Chile en particular.

Se ha dicho que investigar en la Web es algo único, en el sentido de que por un lado presenta desafíos algorítmicos, matemáticos y de sistemas formidables, y por otro puede tener un impacto inmediato en la sociedad. Como alguien que comenzó su carrera en algoritmos, ¿qué es lo que te atrajo, y qué es lo que te atrae ahora, de investigar en la Web? ¿Los temas de investigación han cambiado con el tiempo o son los mismos? ¿Sigue estando el foco en lo algorítmico o se ha movido a lo estadístico? ¿Cuáles son los desafíos más importantes en investigación en la Web para los próximos años?

La Web es actualmente el conjunto de datos más grandes que ha producido la humanidad, pues en la práctica uno puede generar un número infinito de páginas a través de páginas dinámicas. En septiembre de 2008 ya había más de 180 millones de servidores Web y posiblemente más de 20 mil millones de páginas estáticas. Su complejidad se debe a su volumen, continuo cambio y diversidad. Lo que me atrajo inicialmente fue la posibilidad de que las tecnologías de búsqueda que me interesaban pudieran ser usadas por muchas personas, por ejemplo a través del buscador de todo Chile, TodoCL. En otras palabras combinar la teoría y sus aplicaciones. Los temas relacionados con búsqueda de información siguen existiendo pero aparecen otros nuevos y uno de ellos es entender mejor la Web, donde la estadística y la minería de datos son importantes. Pero eso no significa que los temas anteriores dejen de ser importantes. Por eso los desafíos actuales son una mezcla de retos ya conocidos (recolectar la Web, jerarquizar páginas, etc.) y entender mejor la Web para poder aprovechar el conocimiento implícito que las personas generan, ya sea aportando contenido (en particular vía Web 2.0) o usando la Web.

Desde 2002 dirigiste el Núcleo Milenio Centro de Investigación de la Web, el único núcleo Milenio en computación en Chile, y en 2006 abriste el laboratorio de Yahoo! Research en Chile. ¿En qué crees que estas iniciativas han impactado o esperas que impacten en el país. ¿A nivel científico, social, educacional? ¿Ves diferencia entre el estado de la investigación en la Web antes de 2002 y ahora? ¿En qué sentido dirías que se ha avanzado?

A nivel científico el CIW permitió crear el grupo más grande de investigación en computación en un tema específico, sobrepasando la masa crítica. Este grupo ha logrado impacto internacional y por supuesto nacional. Por ejemplo, cuando

Gary Marchionini, conocido investigador de la Universidad de Carolina del Norte, nos visitó hace algunos años, me comentó que él pensaba que teníamos uno de los mejores grupos del mundo. De hecho, la producción de artículos actualmente supera varias veces a la de 2002. Un hito importante para hacer investigación de punta en algunos temas fue el buscador de Chile, TodoCL, que comencé el año 2000,

y permitió tener datos que pocos grupos de investigación académicos tenían. Todos estos hechos fueron fundamentales para convencer a Yahoo! de tener una sede de investigación en Chile, siendo el principal el capital humano.

A nivel social hemos tenido impacto a través de iniciativas que socializan la investigación como la Ventana Digital, los estudios de la Web Chilena y recientemente con el libro Cómo Funciona la Web. A nivel educacional hemos introducido temas nuevos que han permitido formar recursos humanos capacitados en temas diversos como tecnología de buscadores, algoritmos de compresión o Web semántica. A nivel gubernamental el laboratorio de Yahoo! se usa permanentemente como ejemplo para atraer otras iniciativas similares.

¿Qué fortalezas y debilidades tiene montar un laboratorio como el de Yahoo! Chile? ¿Cómo lo compararías con la situación en Barcelona o en EE.UU.? ¿Cuáles son los desafíos para la investigación en computación en Chile, qué hay que perseguir para estar al nivel del primer mundo? ¿La falta de masa crítica es un problema serio?

Chile está lejano del resto del mundo y es difícil traer investigadores de otros países. Tampoco es un país grande que pueda tener una fuente de talento local de gran tamaño. Este es el problema principal y la falta de masa crítica es siempre un problema importante. Los desafíos para hacer una buena investigación son similares a los de un país desarrollado, con el agravante que a veces la infraestructura y otros recursos no están disponibles a los niveles adecuados. Sin embargo estos obstáculos se pueden vencer con esfuerzo y tesón, algo que el DCC ha hecho a lo largo de más de 25 años.

¿Existe alguna particularidad especial en una Web regional, como la chilena o la latinoamericana, que valga la pena intentar explotar o comprender? En el caso de los buscadores, ¿se puede hacer mucho mejor que lo que haría un buen buscador horizontal? ¿De qué manera el estudio de una Web puede revelar información sobre la sociedad subyacente?

Las Web regionales tienen algunas características que son locales, como el idioma y el contenido relacionado con la cultura local. Incluso en el tema de los buscadores se pueden hacer algunas cosas mejor como recorrer la Web en forma más completa y más rápido al estar más cerca y hacer mejor ranking de los resultados al ser la colección más homogénea.

Los estudios de la Web local, además de servir para ver su evolución, pueden ser usados para muchos fines, pues la Web es un reflejo de la sociedad y tiene información social y económica de un país. Con herramientas de extracción de información uno podría mejorar la Web (por ejemplo creando nuevas páginas en forma automática en base a contenido existente). También mediante minería de uso en un buscador regional uno podría recolectar el conocimiento implícito que existe en ese uso, lo que hoy se llama sabiduría de la gente. En particular aprovechar la Web 2.0 (anotaciones, opiniones, etc.) para extender la Web, que sería para mi el inicio de la Web 3.0.

¿Cómo defines la Web 3.0?

Hoy hay diferentes definiciones según el autor, muchas de ellas disponibles en Wikipedia. En mi opinión, la Web 3.0 va a suponer el aprovechamiento del contenido y el uso que está generando la Web 2.0 para distintos objetivos, desde extracción de conocimiento hasta generación automática de contenidos. Todo esto será posible gracias a la minería web, que permite capturar la experiencia y el conocimiento de la gente y ponerlo al servicio de todos, consiguiendo al final una Web mayor y mejor. BITS

Entrevista Internacional

Peter Buneman

Por Pablo Barceló

Peter Buneman es Professor of Database Systems en la School of Informatics de la University of Edinburgh. Su trabajo en ciencia de la computación se ha enfocado principalmente en bases de datos y lenguajes de programación; más específicamente en bases de datos activas, semántica de bases de datos, información aproximada, lenguajes de consulta, tipos para bases de datos, integración de datos, bioinformática e información semisestructurada. Últimamente Buneman ha trabajado en problemas asociados a bases de datos científicas tales como procedencia de datos, archivaciones y anotaciones. También ha participado en numerosos comités de programa y ha sido el chair de ACM SIGMOD, ACM PODS y ICDT. Es además fellow de la Royal Society de Edinburgo, fellow de la ACM, y ganador del Royal Society Wolfson Merit Award. Actualmente se desempeña como director de investigación del UK Digital Curation Centre.

Peter, tu realizaste uno de los primeros trabajos acerca de los fundamentos teóricos de XML y, en particular, acerca del diseño de lenguajes de consulta para XML, durante la segunda mitad de los años '90s. ¿Qué fue lo que te hizo trabajar en XML en ese momento?

Esa es una muy buena pregunta. Antes de trabajar con XML estuve investigando lenguajes de consulta para objetos complejos. Y claro, muchos de estos lenguajes de consulta parecían ser apropiados para formatos de datos científicos en los cuales nosotros estábamos particularmente interesados. Teníamos muchos de estos formatos científicos, y cada uno tenía sus particularidades y eran todos realmente muy interesantes. Desarrollamos entonces algunas álgebras para objetos complejos que se comportaban muy bien como lenguajes de consultas para estos formatos. Pero había uno de estos formatos, en particular un formato para datos biológicos llamado SDBH, que no podía ser tratado con nuestras álgebras. Los datos en este formato se estructuraban como un árbol con algunas propiedades interesantes: el documento contenía muchos nulos (información desconocida o faltante), tenía cierta estructura pero no tenía esquema, entre otras. Esto muestra que antes de interesarme en XML, incluso antes de saber que XML existía, me interesé en el tema de la información semiestructurada. De hecho, personalmente no me considero una persona que haya aportado demasiado al desarrollo de XML, pero sí me gustaría pensar que mi trabajo en información semiestructurada motivó la posterior investigación en lenguajes de consulta para XML.

En el último tiempo hemos vivido una proliferación de diferentes modelos de datos: desde sólo tener el modelo tradicional relacional, hemos pasado en poco tiempo a codearnos con XML, RDF, datos biológicos, etc. Es más, hace unos días me comentabas que los datos utilizados por los lingüistas no se ajustan a ninguno de estos modelos y que, en realidad, corresponden a otro modelo aun inexplorado. ¿Crees que en el futuro veremos una aún mayor proliferación de modelos de datos, o en algún momento se producirá una estabilización en la que unos pocos modelos de datos serán los que predominen?

Creo que de alguna forma este proceso se estabilizará. Primero que todo, aún la gente que trabaja en XML lo hace bajo ciertas simplificaciones del modelo. Por ejemplo, es usual que los DTDs en la práctica sean bastante más simples que los que estudiamos en teoría, es decir, una simplificación usual es que no contengan recursión en vez de ser gramáticas libres de contexto arbitrarias. Y cuando veo estas simplificaciones me da cada vez más la impresión de que XML se parece mucho a algunos modelos de datos muy simples como son las listas, las tuplas, etc. Y en realidad, bajo la mayor parte de estas simplificaciones, la información ya no se puede considerar semiestructurada; de hecho corresponde a una descripción perfectamente estructurada. Pero por otro lado, siempre hay algún elemento semiestructurado en XML: Uno puede agregar arcos al documento, agregar atributos, etc. Y en esa dirección creo que deberíamos empezar a estudiar la relación entre bases de datos y ontologías ¡aunque odio decir esta última palabra!. Creo que habrá un interesante desarrollo ahí. Entonces lo que va a reaparecer - aunque ya están reapareciendo implícitamente - son los modelos más tradicionales de datos, es decir, modelos estructurados, pero esta vez en conexión con las ontologías, que son mucho más libres y representan al elemento semiestructurado.

¿Crees que nuestra función, como "científicos de los datos o de la información," es tratar de entender cada uno de estos modelos de datos por separado, o crees que quizás necesitamos de una teoría de los datos más

general, que englobe a los diferentes modelos de datos que conocemos, es decir, una teoría general sobre los modelos de datos?

Es una muy buena pregunta a la cual creo no tener respuesta. Tú sabes que la gente ha descubierto las muy hermosas relaciones que existen, por ejemplo, entre el modelo relacional y la lógica de primer orden, o entre la información semiestructurada y el área de autómatas. Pero por otra parte existen otros modelos de datos, como los arreglos, que no calzan en este tipo de caracterizaciones. También los streams, donde hay muy importantes conexiones con el área de lenguajes de programación, pero a los cuales prácticamente no hemos estudiado desde el punto de vista de bases de datos.

Pero respondiendo a tu pregunta, creo que nunca tendremos una especie de gran teoría unificadora de qué son los datos. Pero creo que sí vamos a ser capaces de establecer cada vez más conexiones entre los diferentes modelos, y construir mejores lenguajes de consulta para éstos.

Hace unos días me comentabas que una de las cosas que más te gustaba del área de bases de datos es que aún la teoría y la práctica se mantenían relativamente cercanas. Sería bueno si pudieras explicarme un poco más acerca de esa idea.

Bueno, esto es lo que a mí me gusta de la ciencia de la computación, y muy particularmente del área de bases de datos. Siempre me ha gustado mirar ambos lados, teoría y práctica. Y creo que la mayor parte del tiempo la teoría y la práctica de las bases de datos colaboran bastante bien. En ese sentido el estudio de las bases de datos es un tema muy interesante, pues una buena idea teórica puede llegar a tener un impacto práctico rápidamente.

En ese sentido me gustaría mencionar nuestro trabajo acerca del problema de la procedencia de los datos (provenance). Este es un problema que apareció de nuestro estudio, a través de varios años, de qué significaba para los administradores de muchos tipos de datos diferentes entender de dónde provenía su información. Nos dimos cuenta que este problema necesitaba un modelo teórico simple, que permitiera después trabajar con él. Y esto es lo que más me gusta de ese modelo.

Me preocupa un poco si la computación podrá mantener esta dualidad teoría/práctica indefinidamente. Creo que podría llegar a pasar que nuestra disciplina se escindiera en dos diferentes áreas: Por un lado la ingeniería computacional – como un símil de lo que es hoy por hoy la ingeniería mecánica – y por otro lado, totalmente separada, la ciencia de la computación – de la misma forma que la física teórica está totalmente separada hoy de la ingeniería mecánica.

¿Crees por tanto que la ciencia de la computación, en general, y la teoría de bases de datos, en particular, todavía pueden ser "útiles" ?

Claro que sí. El proceso de tratar de "entender" algo es siempre muy interesante y puede llegar a tener impacto en el mundo real. Creo que esto es cierto respecto a mucha de la investigación teórica en ciencia de la computación. Y si no llega a tener un impacto teórico al menos ayuda a dar una visión más completa del problema, o a dirigir futuras investigaciones que sí podrían tener un impacto.

La verdad es que nunca se sabe que tendrá impacto o que no. Y algunas cosas tienen impacto muchos años después. Por ejemplo, yo entré a la comunidad de bases de datos un poco después de la invención de las bases de datos relacionales. Y lo que la gente decía por ese entonces era algo así como "sí, el modelo relacional es muy elegante; pero la verdad es que nunca tendrá impacto en la práctica!" Y la verdad es que la teoría en este caso fue muy importante para poder poner el modelo relacional en práctica. Este tipo de ejemplos ha sucedido también en muchas otras áreas de la computación como por ejemplo en lenguajes de programación, donde ciertas ideas, como la teoría de tipos, han probado ser aplicables muchos años después de su invención.

Por lo demás, la teoría de la computación es barata. Lo que invertimos en ella es ínfimo con respecto a lo que invertimos en los grandes proyectos de software, que además raramente tienen impacto. Además la teoría tiene sus propios medios de auto-regularse, por lo que creo que deberíamos apoyarla. Por supuesto que existirá un efecto colateral, el que encontraremos también en investigación que tiene muy poca posibilidad de ser aplicada y que más bien tiene valor matemático. Pero personalmente no veo que este sea una buena argumentación en contra de este tipo de investigación.

Y por último, ¿podrías contarnos cuáles son a tu parecer las cuatro o cinco contribuciones más importantes de la ciencia de la computación?

Es una pregunta bastante difícil ... Lo que está de alguna forma mas cerca de mi corazón son todas aquellas ideas desarrolladas con relación a los lenguajes de programación: ideas acerca de tipos, de concurrencia, etc. Todas ellas muy hermosas y que han llegado a tener impacto en la práctica, aunque probablemente no en la forma en que se pensó al inicio. Y lo mismo acerca de las bases de datos: Estoy pensando aquí en todos esas elegantes conexiones que se han establecido entre la lógica y los lenguajes de consulta y que han tenido un profundo impacto en las aplicaciones. Es sólo cosa de ponerse a pensar un poco en cómo esto ha influenciado nuestra manera de almacenar y manipular datos. Y puede ser que estas relaciones teóricas y sus aplicaciones aún parezcan un poco difíciles para el usuario, pero es increíble al menos ver como se han simplificado las cosas con respecto a 20 años atrás.

Otras áreas que encuentro fascinantes son la criptografía y la teoría de complejidad computacional. Estas son áreas de las que sé bastante poco, pero que han tenido real impacto en nuestra manera de entender la computación.

Y por último, con respecto a cuáles serán los mayores desafíos en el futuro, me parece que el principal es que la Ley de Moore no podrá ser validada en un sólo procesador para siempre. Creo que, por tanto, nuestro campo tendrá que abrirse a estudiar en profundidad los modelos de computación paralela. Y mi visión es que habrá desarrollos muy interesantes relacionados con esto. BITS

Imagen: Javier Velasco.

Análisis de Redes Sociales: Un Tutorial

MOTIVACIÓN

El *análisis de redes sociales* o *social network análisis (SNA)* es un área de investigación que estudia las redes sociales como grafos, en un intento por hacer sociología de forma precisa y explicar la macrosociología a partir de la microsociología. Y lo está logrando gracias al trabajo conjunto de innumerables investigadores de diversas áreas, entre las que se cuenta la ciencia de la computación.

Una *red social* es un conjunto de *actores vinculados* entre sí. Con esto hemos definido un grafo, pues tenemos los vértices y las aristas o, como le llaman los sociólogos, *actores* y *vínculos*. Los actores pueden ser personas o grupos de éstas: empresas, comunidades, organizaciones de apoyo social, países, ciudades, etc. Los vínculos son cualquier cosa que relacione a los actores, por ejemplo: amor, poder, alianzas, amistad, parentesco familiar, contacto por correo elctrónico, creencias religiosas comunes, rivalidad, etc. (Por ejemplo, vea la figura 1.) Naturalmente, los vínculos pueden ser aristas o arcos (con dirección), y pueden tener uno o más pesos. Además, los grafos resultantes o *sociogramas* pueden llegar a ser más complicados con los grafos clásicos.

Históricamente, el análisis de redes sociales aparece como una de las primeras disciplinas en usar la teoría de grafos para hacer ciencia fuera de las matemáticas [11]. Todo esto parte en los años 30's, luego de que se estableciera la necesidad de hacer las ciencias sociales algo más formal, y así apareciese la *sociometría*. Una de sus líneas usó la estadística para estudiar poblaciones (nivel macro). Otra usó la teoría de grafos para modelar las relaciones entre personas (nivel micro), siguiendo la idea de los árboles genealógicos de la antropología, pero siendo flexibles en lo de genealogía y árboles.

Mauricio Monsalve
Estudiante de Magíster en Ciencias mención Computación, DCC de la Universidad de Chile.
Ingeniero Civil en Computación, de la misma Universidad.
mnmonsal@dcc.uchile.cl

Haciendo breve el asunto, el análisis de redes sociales comenzó creciendo lentamente, a veces a grandes saltos (ver [9, 12]), y demostrando cosas bien peculiares en sociología de las organizaciones, etnias y contactos sexuales [6]. Y ahora ha invadido la investigación sobre la Web y de cómo se hace ciencia [3].

¿Por qué estudiar o investigar redes sociales? Hay muchas razones, entre las cuales podemos considerar el interés por:

1. El estudio de La Web, que concierne directamente a la gente de ciencia de la computación. El análisis de redes sociales ha permitido descubrir propiedades de la Web, como el *Efecto Mateo* (ver más adelate) en la distribución de los vínculos. En una línea similar, veremos la relación entre PageRank [5] y el análisis de redes sociales.
2. El estudio de cómo se hace y mide la ciencia *(scientometrics, epistemometria),* que es de interés para gran parte de la comunidad científica.
3. Las ciencias sociales, ya que el análisis de redes sociales se utiliza activamente en sociología, antropología, ciencia política, gestión organizacional, medios de comunicación *(social media analysis),* etc.
4. La teoría de grafos y la matemática discreta, el fundamento técnico del análisis de redes sociales.
5. El modelamiento, la simulación y el diseño de algoritmos, habilidades y conocimientos claves que han hecho que el análisis de redes sociales escale en tamaño.

En esta exposición se presentan varias oportunidades específicas de investigación en el análisis de redes sociales.

¿CÓMO ESTUDIAMOS UNA RED SOCIAL?

Primero necesitamos recopilar información fiable y expresarla como un grafo o sociograma. Luego, analizamos el grafo para determinar propiedades de la red social original. Aquí veremos cuatro maneras de analizar esta información.

Estudiando las características generales

Es posible que existan muchos tipos de redes por ahí que faltan ser clasificadas, pero ya se observan fenómenos bien frecuentes en ellas.

Redes de mundo pequeño (small world networks). ¿Quién no ha escuchado hablar de los *seis grados de separación* o que estamos a seis personas de distancia de todo el mundo? Este fenómeno se conoce como *mundo pequeño* (small world) y ocurre en las redes que tienen una conectividad especial que hace que la distancia promedio, entre dos actores cualquiera, sea muy pequeña en comparación con el tamaño (número de actores) de la red.

Stanley Milgram, un importante psicólogo norteamericano, realizó un experimento que medía la distancia promedio entre personas, en redes de contacto [9, 12]. Eligió personas de Ohama (Nebraska) y Wichita (Kansas) para que se contactaran con personas de Boston (Massachusetts). La gente de Ohama y Wichita indicaba sí conocían, o no, a las personas de Boston. En caso contrario, remitían a un contacto que pudiera conocerlas, con las que se repetía el proceso. Milgram, informado de todo esto, pudo medir cuál era el largo de los caminos recorridos. El resultado: 6 personas de distancia en promedio.

Milgram realizó muchos otros experimentos de conectividad. Muchos criticaron sus procedimientos, calificándolos como mitos urbanos, incluso recientemente [2]. Sin embargo, esta propiedad se sigue observando una y otra vez [3]. Su aparición es tan recurrente que se considera conocimiento general, e incluso es parte de teorías de innovación [13] y propuestas de abandono a los medios de comunicación masivos [10].

Redes libres de escala (scale free networks). Muchas redes, como las de citación de artículos científicos, la Web, y muchas otras, tienen una distribución de grados que sigue una ley de potencias o similar [3]. A esas redes las llamamos *redes libres de escala (scale free networks),* porque si tomamos un subgrafo de esta red lo más probable es que los grados se sigan distribuyendo como ley de potencias.

Las redes libres de escala son interesantes porque son regidas por leyes de potencia, que se repiten en otros casos como en la distribución del ingreso; *el rico se vuelve más rico mientras el pobre se hace más pobre*, efecto tan típico que hasta aparece en la Biblia. En efecto, se le llama Efecto Mateo, y es reconocido en sociología, economía y comunicaciones, tal como lo indica el famoso sociológo estructuralista Robert Merton [8].

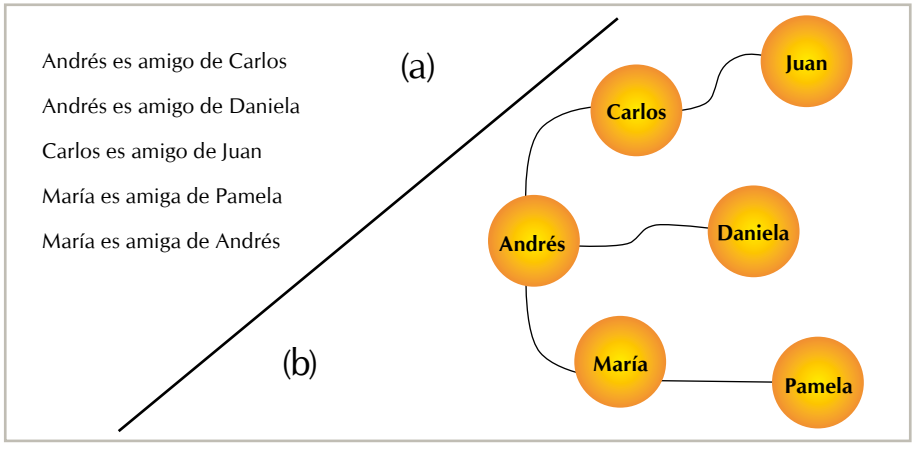


Fig. 1 Especificando una red social.

Red social que modela una situación de amistades. Cada (a) relación de amistad se traduce en un (b) vínculo en el grafo. Los vínculos no tienen dirección, porque la amistad es una relación simétrica o refleja (A es amigo/a de B⇔B es amigo/a de A).

Estudiando la posición de los actores

El concepto tras la posición o localidad de un actor en una red corresponde al acceso que tiene al resto de la red. En principio, sabemos que dos actores ocupan el mismo lugar en la red si comparten los mismos vecinos (*equivalencia estructural*, una versión local de *isomorfismo* de vértices). Pero en general deseamos ir más lejos. En esta necesidad definimos las medidas de *centralidad*, que *miden* la posición de un actor en una red de acuerdo a ciertos criterios.

Centralidad de grado (degree centrality). Un hombre o mujer popular es aquel que tiene muchos amigos o conocidos, ¿no? Con esta simple intuición, definimos nuestra primera medida de centralidad: la centralidad de grado (degree centrality).

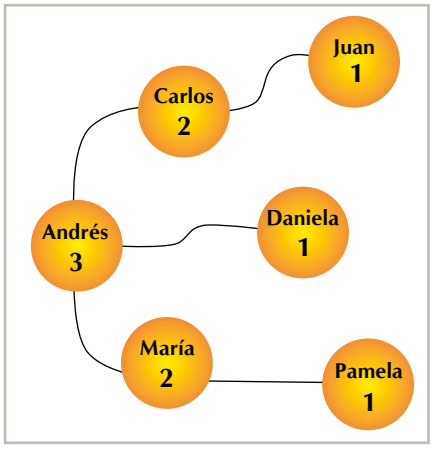


Fig. 2 Centralidad de grado.
La centralidad de cada actor se calcula como su número de vecinos.

En términos de grafos, la centralidad de grado de un actor se calcula como su número de vecinos. Si estamos modelando una red social de amigos, la centralidad de cada actor consiste en su número de amigos. Un buen ejemplo se puede apreciar en la figura 2.

Centralidad c(β) de Bonacich (Bonacich's c(β)). Hay gente cuya favorable posición en una red social les permite iniciar procesos influenciales como la transmisión de

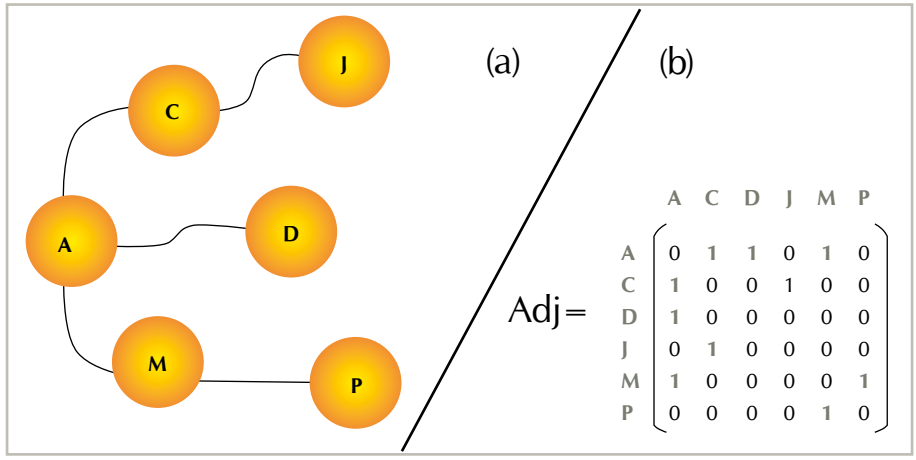


Fig. 3 Grafo y su matriz de adjacencia.
Como vemos, los actores están asociados a las filas y a las columnas de la matriz de adjacencia. Si hay un vínculo entre el i-ésimo y el j-èsimo actor, entonces la componente i,j de la matriz de adjacencia será 1. De lo contrario, será 0.

creencias, chismes (*gossip*), publicidad viral, etc. En estos casos, el proceso empieza en un actor y se distribuye a su vecinos, los cuales redistribuyen a sus propios vecinos, sucesivamente. Entonces, podemos proponer una medida de centralidad para tales situaciones, que consista en contar los caminos.

Sin embargo, las redes con ciclos nos dan problemas pues tienen infinitos caminos. Por eso, no podemos contar los caminos así nada más. La solución práctica a este dilema es *atenuar* los caminos usando una *tasa de descuento*, tal como se usa en la evaluación económica, las series de potencias, etc. Así, los caminos más largos se suman como números más pequeños, y los infinitos son cero.

Usar una tasa de descento que hace más pequeños los caminos más grandes tiene ventajas conceptuales. Los procesos influenciales como los chismes, las creencias, etc. pueden ser muy efectivos en distancias cortas, pero su difusión se hace menos efectiva (o lenta) a grandes distancias. Ajustando la tasa de descuento se puede simular cuán rápido se atenúa un proceso de difusión.

Pasando a lo matemático, definimos la centralidad $c(\beta)$ de Bonacich como $c(\beta) = (\sum_{k\geq 1} \beta^k A^k) \cdot \vec{1}$, donde β es la tasa de descuento y A^k cuenta los caminos de largo k entre cualquier par de actores. Esta es una propiedad de la matriz de adjacencia, la cual especifica al grafo como una matriz (vea la figura 3). Esta centralidad ha sido llamada centralidad $c(\beta)$ de Bonacich por su creador. Sin embargo, la idea es bastante antigua, casi tanto como el análisis de redes sociales.

Un ejemplo práctico de la centralidad $c(\beta)$ se muestra en la figura 4, en la cual se ilustra la función $c(\beta)$ evaluada en 0,5.

Notemos que la serie $\sum_{k\geq 1} \beta^k A^k$ converge (si lo hace) a $\beta A(1-\beta A)^{-1}$, donde I es la matriz identidad, así que podemos calcular $c(\beta)$ como $c(\beta) = \beta A(1-\beta A)^{-1} \vec{1}$.

¿Qué valor de β usar? Obviamente, un b menor que uno, pero eso no garantiza convergencia. La solución que asegura la convergencia es usar un β menor que el valor propio de A que tiene mayor norma (recordemos álgebra lineal). Un método para calcular el vector y el valor propio más grandes se presenta en el siguiente punto.

Centralidad de vector propio (eigenvector centrality). ¿Qué tal si definimos la centralidad de manera recursiva? Digamos que un actor central es aquel que tiene un vecindario con buena centralidad. Por ejemplo, un feriante puede conocer a mucha gente común mientras que un político puede

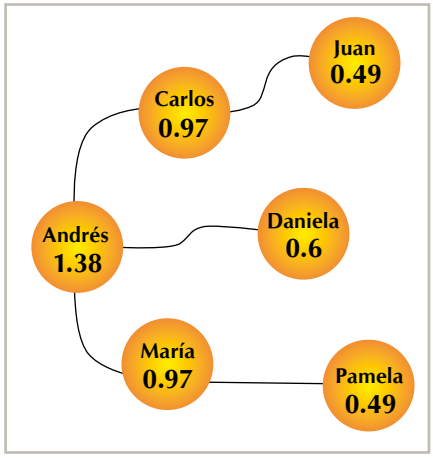


Fig. 4 Centralidad c(0;5).
En el grafo, se ve la centralidad c(β) para cada actor, evaluado con β = 0;5. Se puede ver cómo la centralidad de un actor es influenciada por los vecinos.

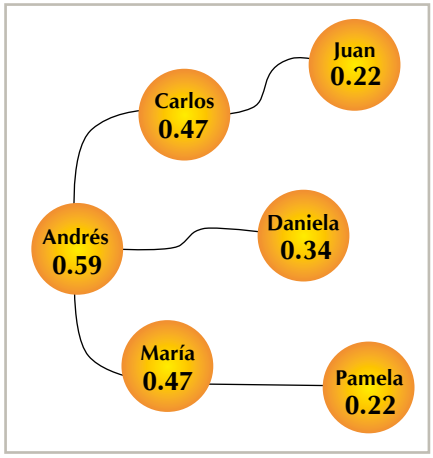


Fig. 5 Centralidad de vector propio.
En el grafo,se ve la centralidad de vector propio,calculada con $\vec{c}=(A^{50}\vec{1})/||A^{50}\vec{1}||$. *Se puede ver cómo la centralidad de unactor es influenciada por sus vecinos.*

conocer menos gente, pero que son personas influyentes. Al final, el político está mejor posicionado en influencia que el feriante, aunque conozca menos gente (¡pero no conoce poca!). Ahora, no todo el vecindario de un actor contará con la misma centralidad, por lo que hay que considerar que los actores con mayor centralidad son más influyentes que el resto.

Siguiendo el concepto de centralidad recursiva, diremos que la centralidad de un actor es *proporcional* a la suma de las centralidades de sus vecinos en el grafo. Matemáticamente, esto se expresa como $c_i=\lambda\sum_j a_{ij}c_j$, donde λ es la constante de proporcionalidad y a_{ij} es el elemento de la fila *i* y la columna *j* de la matriz de adjacencia *A* de nuestra red social. En álgebra lineal, $\vec{c}=\lambda A\vec{c}$, o sea, $\vec{c}$ es un vector propio de *A* (y λ^{-1} es su correspondiente valor propio). ¡Ahora queda claro el nombre de la centralidad de vector propio!

Pero el lector experto notará que, desafortunadamente, este problema no tiene solución única; en general, una matriz tiene varios vectores propios diferentes. ¿Cuál usar? Cualquier valor propio positivo tiene sentido en la ecuación, ¿no? Bueno, los investigadores decidieron dejar todo en el valor propio más alto, por una propiedad muy sencilla: su vector propio asociado es fácil de calcular. La sucesión $\vec{c}_{k+1}=A\vec{c}_k/||A\vec{c}_k||$, que inicia $\vec{c}_0=\vec{1}$, nos permite obtener rápidamente este valor.

De manera más sencilla, podemos aproximar la centralidad de vector propio como $\vec{c}\approx A^k\vec{1}/||A^k\vec{1}||$, para un *k* adecuado tal que la aproximación varíe poco (para *k* y *k*+1 hay diferencias minúsculas). Esto se puede asegurar con un valor de *k* grande (por ejemplo, 50). Un ejemplo práctico que usa esta aproximación se ve en la figura 5.

Centralidad de cercanía (closeness centrality). Otra medida de posición sale de considerar la distancia promedio al resto de la red. El actor que está más cerca de todo otro elemento de la red es el más central, versa la idea tras la centralidad de cercanía *(closeness centrality)*. Pero nosotros no medimos "cercanías", sino "distancias"; o sea, lo contrario.

Matemáticamente podemos expresar la centralidad de cercanía como el inverso a la suma de las distancias, o sea, $ci=\frac{1}{\sum_j d_{ij}}$, donde d_{ij} es la distancia entre el actor *i* y el actor *j*. Obviamente hablamos de distancias eucídeas o rutas mínimas.

Centralidad de intermediación (betweenness centrality). Digamos que un grupo terrorista se toma Linares (¡en Linares no pasa nada!) e impide el movimiento de camiones entre las zonas central y sur del país. ¿Qué hacen? Desconectan a Chile. Linares, por inocente y tranquilo que parezca, es una ciudad clave en la red de suministro de Chile, pues es camino obligado. O sea, todos los caminos pasan por Linares. (La realidad es que es fácil hacerle el quite, pero éste es un ejemplo.) Esto nos inspira a repensar las medidas de centralidad.

En estrategias militares y terrorismo, es importante distinguir los actores claves. Si son atacados, desconectan una red, o interrumpen sustancialmente los flujos que se pudieran producir en ésta. Estos actores, que son objeto de ataque y defensa, se pueden descubrir contando cuántas rutas mínimas pasan por ellos; o sea, por su calidad de intermediaros o puntos intermedios. Por eso definimos la centralidad de intermediación como el número de rutas mínimas en las que el actor participa.

Estudiando los grupos que tiene

La detección de comunidades, grupos, cliques (grupos exclusivos), etc. es tema de alto interés en redes sociales. El asunto es complicado pues no es fácil definir un grupo. La definición es fácil cuando hablamos de una estructura formal, cuando existe un grupo definido y un grupo de adherentes que dice ser parte del grupo. Por ejemplo, Chile y los chilenos. Pero todo se vuelve complicado, oscuro, hasta esotérico cuando hablamos de la estructura *informal*. Un grupo de amigos es un montón de gente que son todos o casi todos amigos entre sí, pero ellos a su vez tienen varios amigos comunes... ¿Cuáles pertenecen al grupo y cuales no?

Técnicas de detección. Técnicas para detectar grupos hay muchas; hay muchos algoritmos, con muchas velocidades diferentes, que obedecen a diferentes ideas de cómo se detecta un grupo, situación muy opuesta a la de las medidas de centralidad.

Una manera tradicional consiste en reducir la detección de grupos a una clasificación o *clustering*. Dentro de estas técnicas están el

tradicional k-Means, los algoritmos genéticos, el análisis de *modularidad* (el número de vínculos entre grupos es pequeño, dentro de grupos es alto), etc. Adicionalmente, estas técnicas son parametrizables (i.e. número de clases en k-Means, modularidad mínima, etc.), lo que permite analizar la calidad de la clasificación. Aquí se hace posible usar *árboles jerárquicos* para decidir cuándo la clasificación es *buena*.

Otra manera tradicional consiste en ver el problema como uno de teoría de grafos. Por ejemplo, la *coloración* es una forma tradicional de hacer clasificación en grafos. En este caso, también es posible ver el problema como uno de *equivalencia estructural* transformado a uno de *equivalencia regular*: "en un grupo de amigos, los amigos compartimos los mismos amigos" (esto define un algoritmo iterativo). Adicionalmente, se pueden buscar *cliques y k-Cliques* para encontrar los grupos.

Maneras más novedosas incluyen el uso de las medidas de centralidad: "en un grupo, todos los actores son cercanos" (centralidad de cercanía), "un grupo es una red más o menos aislada del resto" (centralidad de intermediación). También se incluyen medidas *democráticas*, que consisten en consensuar dos o más criterios diferentes de detección de grupos.

Una guía a la historia de los algoritmos de clustering la hace Freeman [4]. Otra revisión más sintética la entrega Boccaletti [3].

Los problemas. Aún queda mucha investigación por hacer en el tema de detección de comunidades. Aquí listo algunos de esos desafíos.

Entre los problemas conceptuales, nos encontramos con: ¿Qué es un grupo o una comunidad? El caso formal es sencillo, pues los actores declaran su pertenencia a un grupo, pero el caso informal es bastante complejo. Asimismo, tenemos el siguiente problema: ¿Cuándo es conveniente comparar con un caso formal? También debemos considerar que los actores pueden pertenecer a varios grupos, cosa que muchos algoritmos no admiten.

Entre los problemas de eficiencia (rapidez), notamos que hay muchos algoritmos que son NP-HARD por tratar de cumplir exactamente una definición, lo que motiva a usar aproximaciones. Pero debemos ser aún más exigentes: si hablamos de cientos, miles o millones de vértices, un algoritmo de orden Q(n3) puede ser desastrosamente lento. ¡No podemos conformarnos sólo con estar en la clase P (tiempo polinomial)! Las redes son cada vez más grandes, y la necesidad de algoritmos más rápidos es cada vez mayor.

Como indicamos, se hace necesario aproximar en muchos casos. Sin embargo, esto supone nuevos desafíos: ¿Cuándo es bueno usar algoritmos aleatorios? ¿Cuándo es bueno usar heurísticas? Más aun, debemos tener en cuenta que pueden haber actores y vínculos que no consideramos en la construcción del grafo, por lo que nuestros algortimos deben ser precisos incluso cuando la información escacea o falla.

Personalmente tengo las siguientes interrogantes, las que, de ser contestadas, podrían dar origen a varias publicaciones:

1. ¿Puede un grupo contener otros grupos? Esto ocurre en las estructuras sociales formales.
2. ¿Cómo generar algoritmos de clustering para nuevas representaciones gráficas?
3. ¿Cómo podemos definir clusters cuando hay grafos dirigidos? ¿Qué ocurre en el caso de grafos con pesos?
4. Si consideramos un algoritmo iterativo, ¿podemos usar un resultado aproximado para generar otro más preciso? (Combinar algoritmos.)

Visualización

La visualización de las redes sociales también sirve como método para descubrir propiedades de ésta, aunque tiene menos peso teórico en el análisis. Pero cuenta con la ventaja de alimentar rápidamente la intuición del investigador.

Visualizar redes complejas es un gran desafío; por lo general, se busca presentar gran cantidad de información de forma

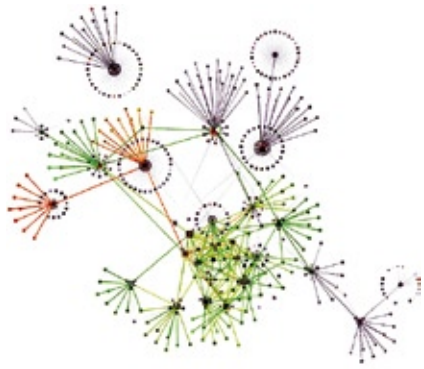


Fig. 6 *Visualización de una red social.*

estética. Se busca la claridad y la simpleza, pese a la gran complejidad de los datos, como se ilustra en el ejemplo de la figura 6. Y hay que considerar que hay muchas potenciales vistas de los datos, que pueden ilustrar propiedades diferentes: centralidad, comunidades, jugadores clave (que, si desaparecen, desconectan la red), etc.

Tal como en la detección de comunidades, existe una gran variedad de algoritmos para visualizar redes sociales. Cada uno obedece a una idea u objetivo diferente, aunque muchas veces se busca la presentación instantánea.

¿CÓMO OBTENEMOS REDES SOCIALES?

Uno de los desafíos con las hipótesis sobre redes sociales es reproducir los fenómenos que se dice que ocurren.

La Web. Una de las redes más estudiadas en computación es la Web, cuya relevancia al área es clarísima. La Web es una red gigante, masiva, en donde participan millones de páginas con vínculos entre sí. Notemos que hay muchos tipos diferentes de páginas web; estáticas y dinámicas (que se generan en el vuelo), que se actualizan, algunas que se borran, otras se crean; hay buscadores con vínculos a grandes porciones de la red, catálogos, sitios de noticias, blogs, foros, sitios de fotografías, de vídeos, bibliotecas, sitios corporativos, etc. los cuales están llenos de páginas y vínculos.

Los desafíos que pone la Web para su estudio caen en los problemas de recuperación y consulta de información. Este es el problema tradicional de los buscadores, que deben buscar y buscar páginas web, siguiendo vínculos, y deben recuperar y clasificar su contenido. Luego, deben explotar la bases de datos construidas para responder las consultas.

Una de las grandes aplicaciones de las medidas de centralidad se da justamente en la red; Google, en vez de buscar páginas por la calidad de su contenido, las busca por su fama. Cada página tiene una nota, un ranking, que sale de la fama de las páginas que la referencian. Este algoritmo, llamado Page Rank, es justamente una aproximación de la *centralidad de vector propio*. Veamos la propia explicación que da Google:

> *(...) En lugar de contar los vínculos directos, Page-Rank interpreta un vínculo de la página A a la B como un voto para la página B por parte de A. (...) Esta tecnología también tiene en cuenta la importancia de cada página que efectúa un voto, dado que los votos de algúnos se consideran de mayor valor, con lo que incrementan el valor de la página a la que enlazan. [5]*

Estudiar la Web es un asunto colosal; la Web es demasiado grande, por eso se hacen estudios locales. Por ejemplo, en Chile se puede explotar el registro de NIC, con lo que se ha realizado el Estudio de la Web Chilena [1].

Sitios de redes sociales, Web 2.0. Muchos de los datos de redes sociales son recopilados de los sitios sociales, sitios extremadamente populares cuya estructura no está completamente predefinida sino que se construye dinámicamente de acuerdo a las acciones de sus usuarios. En estos sitios se suele cumplir con los 5 ó 6 grados de separación de Milgram.

Entre los sitios sociales se cuentan: Facebook, Wikipedia, Fotolog, Habbo, Last.fm, Orkut, Youtube, MySpace, Xing, Flickr, Piccasa, Advogato, SourceForge, MyHeritage, aSmallWorld, Broadcaster.com, Classmates.com, DeviantART,Twitter, Sonico.com, etc. (Más de alguno de estos sitios debiera sonar conocido).

Otros registros digitales. Los seres humanos solemos dejar huellas de nuestras interacciones en los medios digitales, más allá de los sitios sociales. Por ejemplo, foros, news, irc, email, CVS-SVN, comercio electrónico, telefonía IP, etc. son evidencias digitales de interacciones humanas. Todas éstas están sujetas a estudio. Sin embargo, aparecen los dilemas éticos de la información confidencial que los sitios sociales expresamente hacen pública.

Encuestas. Este es el método de recuperación de información social más usado en el estudio de las redes sociales tradicionales que están fuera de la Web. Sin embargo, suele ser muy caro realizar estudios de este tipo, sobre todo en papel. Mas aun, muchas veces se requieren investigadores en terreno que supervisen el correcto proceder de las encuestas. Versiones más baratas son las encuestas por Teléfono e Internet, aunque su validez es limitada.

Simulación. Esta es una técnica muy usada por bastantes científicos sociales que trabajan con comunidades artificiales. Sin embargo, su uso aparece más útil en la comprobación de hipótesis que en el análisis desde cero. Usando simulación se puede responder a una pregunta como "¿este proceso social genera redes con estas características?". Luego, las redes artificiales y las reales se comparan con las técnicas de análisis presentadas previamente, y se puede concluir el alcance de una hipótesis.

CONCLUSIONES

El análisis de redes sociales es un área que presenta muchas oportunidades de investigación para la gente de ciencia de la computación. Vimos los exigentes desafíos algorítmicos que propone el área, las diversas métricas que se obtienen de los grafos, su incidencia en el estudio de La Web y el diseño de buscadores, incluso ligeramente el uso de la simulación en el área (que se vio muy poco para lo extendido que es su uso). El área va mucho más allá de lo que son la informática y computación sociales; da a la computación y la matemática discreta un lugar privilegiado en la teoría social. BITS

REFERENCIAS

[1] R. Baeza-Yates, C. Castillo, E. Graells. "Características de la Web Chilena 2006". Centro de Investigación de la Web. 2006. http://www.ciw.cl/material/web_chilena_2006/index.html

[2] Blastland interviewing Kleinfield. "Connecting with people in six steps". More or Less. BBC News. http://news.bbc.co.uk/1/hi/programmes/more_or_less/5176698.stm

[3] S. Boccaletti et al. "Complex networks: Structure and dynamics". Physics Reports 424, 2006, 175-328.

[4] L. Freeman. "Finding Social Groups: A Meta-Analysis of the Southern Women Data". http://moreno.ss.uci.edu/85.pdf

[5] Información corporativa de Google: tecnología. Visto en Octubre de 2008. http://www.google.cl/corporate/tech.html

[6] E. Lawmann. "A 45-year retrospective on doing networks". Connections 27(1), 65-90. 2006.

[7] "mc-50 map of FlickrLand: flickr's social network". http://www.flickr.com/photos/gustavog/4499404/in/set-113313/

[8] R. Merton. "The Matthew Effect in Science". Science, 159 (3810): 56-63, Enero 5, 1968.

http://www.garfield.library.upenn.edu/merton/matthew1.pdf

[9] S. Milgram. "The Small World Problem". Psychology Today, 1967, Vol. 2, 60-67.

[10] L. De Rossi. "The Power Of Open Participatory Media And Why Mass Media Must Be Abandoned". Robin Good, Master New Media. Visto en Octubre de 2008. http://www.masternewmedia.org/news/2006/03/20/the_power_of_open_participatory.htm

[11] J. Scott. "Social Network Analysis: A Handbook". Sage Publications. 2000.

[12] Travers, Jeffrey, and S. Milgram. "An Experimental Study of the Small World Problem". Sociometry 32, 1969, 425-443.

[13] D. Ward. "Knock, Knock, Knocking on Newton's Door: Building Collaborative Networks for Innovative Problem Solving". Defense AT&L Journal. Marzo-Abril de 2005. http://www.dau.mil/pubs/dam/03_04_2005/war-ma05.pdf

Khipu
Centro para la Investigación de Bases de Datos

Marcelo Arenas
Profesor Asistente, DCC, Pontificia Universidad Católica de Chile. Doctor en Ciencia de la Computación, University of Toronto, Canadá. Licenciado en Matemáticas, Magíster en Ciencias de la Ingeniería e Ingeniero Civil de Industrias mención Computación, Pontificia Universidad Católica de Chile.
marenas@ing.puc.cl

Jorge Pérez
Estudiante de Doctorado en Ciencia de la Computación, Pontificia Universidad Católica de Chile. Ingeniero Civil en Computación y Magíster en Ciencia de la Computación, de la misma Universidad.
jperez@ing.puc.cl

Uno de los primeros pueblos en habitar los Andes sudamericanos, los Incas, desarrollaron y utilizaron un medio sofisticado para almacenar información administrativa, el khipu. Un khipu está formado por piezas de cuerdas que están amarradas de algunas formas particulares, y que implementan un modelo complejo de datos (un khipu se asemeja a una estructura de árbol, diríamos en terminología moderna). El khipukamayuq era el experto en la creación, actualización e interpretación de la información codificada en un khipu, algo así como el actual administrador de una base de datos. Desafortunadamente, los invasores españoles del siglo XV ordenaron la destrucción de los khipu y cazaron a los khipukamayuq, destruyendo temporalmente una rica tradición en el almacenamiento de información.

El almacenamiento y manipulación de datos ya significaba un importante desafío para las antiguas civilizaciones. Hoy en día, las tecnologías digitales de muy bajo costo, permiten almacenar y difundir información en cantidades inimaginables en el pasado. Un ejemplo cotidiano de esto es la Web, que funciona como un gigantesco repositorio de datos. Además, prácticamente todos los grandes proyectos de investigación dedican importantes esfuerzos a la recopilación y análisis de datos. Basta citar ejemplos como el proyecto del Genoma Humano en biología, o proyectos de astronomía como el Observatorio Virtual. Es así como en el mundo moderno el manejo eficiente de información no sólo es un desafío sino más bien una necesidad.

El centro Khipu, inspirado en el trabajo pionero de nuestros antepasados sudamericanos, tiene como objetivo desarrollar, desde nuestra región, investigación de impacto mundial en el área de bases de datos y procesamiento de información. El Khipu

nace en el año 2008 como un esfuerzo conjunto de investigadores nacionales por crear un centro de investigación de excelencia, donde se consoliden los logros ya alcanzados, y se generen las interacciones necesarias para potenciar e incrementar la investigación de alto nivel en nuestro país. Hoy Khipu convoca a profesores, investigadores, alumnos y colaboradores de distintas regiones de Chile.

En este artículo resumimos la concepción de Khipu como centro de investigación en bases de datos y manejo de información, quienes lo integramos, los logros que hemos alcanzado y nuestra visión hacia el futuro.

KHIPU: UN ESFUERZO

A NIVEL NACIONAL

Durante los últimos 10 años se ha producido un incremento sustancial en el número de académicos chilenos que han realizado sus estudios de doctorado en el extranjero en el área de bases de datos. Esto permite tener hoy una masa crítica suficiente para instalar en Chile un centro de investigación en el área, que pueda tener impacto a nivel mundial.

El núcleo de Khipu esta formado por ocho investigadores asociados. Como muestra la siguiente tabla, todos estos investigadores obtuvieron sus doctorados en universidades norteamericanas y europeas. Es importante destacar que estos investigadores no sólo están concentrados en Santiago, haciendo de Khipu un esfuerzo a nivel nacional.

Nombre	Afiliación actual	Estudios de doctorado	País	Año
Marcelo Arenas	PUC Chile	University of Toronto	Canadá	2005
Pablo Barceló	U. de Chile	University of Toronto	Canadá	2006
Loreto Bravo	U. de Concepción	Carleton University	Canadá	2007
Benjamín Bustos	U. de Chile	University of Konstanz	Alemania	2006
Mónica Caniupan	U. del Bío-Bío	Carleton University	Canadá	2007
Claudio Gutiérrez	U. de Chile	University of Wesleyan	USA	1999
Carlos Hurtado	U. Adolfo Ibáñez	University of Toronto	Canadá	2002
Andrea Rodríguez	U. de Concepción	University of Maine	USA	2000

Para nuestro centro es también fundamental la formación de alumnos de pregrado, magíster y doctorado en el área de bases de datos. Durante los últimos años, los investigadores de nuestro centro han participado en la formación de un importante número de alumnos, y en la actualidad contamos con 10 alumnos de magíster y 6 alumnos de doctorado.

DESARROLLANDO INVESTIGACIÓN DE IMPACTO MUNDIAL

Una de las premisas de nuestro grupo es generar conocimiento de relevancia mundial. Por esto, parte de nuestro esfuerzo ha estado enfocado en desarrollar áreas de vanguardia y en publicar nuestros resultados en las conferencias y revistas más prestigiosas en nuestra área, y también en computación en general. De hecho, desde el año 2000 el núcleo de investigadores de Khipu ha publicado más de 160 artículos en conferencias y revistas internacionales.

La investigación que realiza nuestro grupo abarca diversas áreas sobre el procesamiento de información. Algunas de las áreas que se están desarrollado son: bases de datos multimedia, bases de datos geoespaciales,

El centro Khipu, inspirado en el trabajo pionero de nuestros antepasados sudamericanos, tiene como objetivo desarrollar, desde nuestra región, investigación de impacto mundial en el área de bases de datos y procesamiento de información.

integración e intercambio de datos, manejo de datos en la Web Semántica, manejo de información inconsistente. A modo de ejemplo, detallamos algunas de estas líneas de investigación y parte de los logros obtenidos.

- **Manejo de información inconsistente.** Usualmente a la información almacenada en una base de datos se le exige que satisfaga ciertas restricciones de sanidad (por ejemplo, que una persona tenga sólo un número de RUT asociado). En muchas aplicaciones estas restricciones se violan dejando a la base de datos en un estado *inconsistente*. La pregunta entonces es cómo poder razonar sobre datos con este tipo de inconsistencias, de manera eficiente y limpia. Esta línea de investigación fue iniciada por miembros de nuestro grupo quienes desarrollaron los fundamentos teóricos necesarios para estudiar el problema. Hoy diversos grupos de investigación a nivel mundial están desarrollando las distintas aristas del tema basándose en los resultados generados por nuestro grupo, aplicando también los resultados en áreas como bioinformática y sistemas de información geoespaciales.
- **Manejo de datos multimedia.** Las bases de datos multimedia están compuestas por imágenes, gráficos en 2 y 3 dimensiones, audio, y videos digitales, en conjunto con datos de texto. En la actualidad existen diversas aplicaciones que necesitan de datos multimedia, como por ejemplo aplicaciones de manufactura, arte, y cine digital. Una de las áreas desarrolladas por integrantes de nuestro grupo es la *búsqueda* por *similitud* en bases de datos multimedia. Esta técnica permite obtener mejoras considerables en la efectividad de las búsquedas en este tipo de bases de datos. Muchos desafíos existen en esta área, como por ejemplo, el diseño de nuevos algoritmos y estructuras de índices adecuadas para acelerar las búsquedas de multimedia.
- **Manejo de datos en la Web Semántica.** La Web Semántica es una extensión de la Web tradicional, pensada en la inclusión de metadatos que permitan, tanto a personas como a máquinas, entender el *significado* y las *relaciones* de la información almacenada en la Web. Una de las líneas de investigación de nuestro grupo tiene que ver con desarrollar las herramientas teóricas necesarias para poder manipular los metadatos semánticos en la Web. Uno de los principales logros alcanzados en esta área fue la formalización de la semántica y subsiguiente estudio del lenguaje de consulta SPARQL, que hoy es el lenguaje estándar para consultar los metadatos de la Web Semántica.

Parte del impacto de la investigación generada por Khipu se puede medir por los reconocimientos internacionales que han obtenido nuestros miembros. Dos artículos escritos por miembros de Khipu han sido reconocidos con el *Best Paper Award* (premio al mejor artículo) en la conferencia más importante de Teoría de Bases de Datos (*ACM Principles of Database Systems*). Adicionalmente, tres de nuestros artículos han sido reconocidos con el *Best Paper Award* en las conferencias más importantes del área de Web Semántica (*International Semantic Web Conference* y *European Semantic Web Conference*). Nuestros alumnos han obtenido también importantes premios. Uno de ellos se convirtió recientemente en el primer alumno chileno en recibir la *Microsoft Research Fellowship*, el reconocimiento académico más importante que entrega Microsoft a alumnos de doctorado.

NUESTRA VISIÓN DEL FUTURO

Las habilidades para almacenar, procesar y analizar información de manera eficiente son fundamentales para el desarrollo de nuestra sociedad. Es por esto importante contar con centros de investigación que puedan proponer soluciones a estas problemáticas. Chile tiene hoy una oportunidad de contar con uno de estos centros, por el aumento significativo en el número de investigadores en el área de bases de datos que residen en nuestro país. Es por esto que decidimos crear Khipu, un esfuerzo nacional para formar un centro de investigación en el área de bases de datos y manejo de conocimiento. En el futuro cercano, esperamos que Khipu se transforme en un referente latinoamericano en el área, tanto por la investigación en bases de datos, como por la formación de alumnos de pregrado, magíster y doctorado que colaboren con el desarrollo de nuestra región. También esperamos que nuestra investigación continúe teniendo impacto a nivel mundial, y que nuestro centro se convierta en un lugar a visitar al momento de buscar nuevas y mejores tecnologías de bases de datos. BITS

Más información sobre nuestro centro puede ser obtenida en `www.khipu.cl`

CONFERENCIAS:

LA-Web 2008

Por Mari-Carmen Marcos
Universitat Pompeu Fabra
Barcelona-España

Los días 28, 29 y 30 de octubre de 2008 la ciudad de Vila Velha, estado de Espírito Santo de Brasil, fue el punto de encuentro de la última edición de LA-Web, un congreso internacional con sede en Latinoamérica apoyado por el comité de la IW3C2 de la International World Wide Web Conference y que por sexta vez reúne a investigadores de varios continentes.

Vila Velha, unida a Vitória (capital de Espírito Santo) por medio de varios puentes y presidida por un monasterio en lo alto de la colina, nos recibió con clima tropical y una larga playa que sólo algunos, en las últimas horas antes del regreso al aeropuerto, pudimos disfrutar. El comité de organización local, en coordinación con la conferencia WEBMedia 2008, celebrada paralelamente a LA-Web y con la que compartimos algunos ponentes, lo tenía todo preparado para nuestra llegada y estuvo atento a los detalles durante los tres días que duró el evento.

LA-Web se abrió con la conferencia invitada de Manuel Pérez-Quiñones, especialista en Interacción Persona-Ordenador en el Center for Human-Computer Interaction de Virginia Tech, Estados Unidos. Su charla abordó un tema de alta actualidad: ecosistemas personales de información, entendidos como el conjunto de dispositivos con los que cada persona interactúa para satisfacer sus necesidades de información, por ejemplo el notebook, el celular, el iPod, la webcam, etc. El tema resultó muy atractivo y, de hecho, todos nos sentimos identificados con los problemas que el expositor presentó con ejemplos muy acertados. Pérez-Quiñones apuntó a la situación que vivimos hoy en día: utilizamos distintos dispositivos para obtener información e interactuamos con cada uno por separado, sin posibilidad de poder establecer una relación entre ellos. Mientras los diseñadores se centran en imitar las funcionalidades y el aspecto de los nuevos dispositivos con respecto a los ya existentes, Pérez-Quiñones reclamó la necesidad de generar ecosistemas de información donde, al igual que ocurre en los ecosistemas biológicos, se atienda a las relaciones entre los distintos dispositivos para satisfacer las necesidades de información de los usuarios. En su charla presentó múltiples ejemplos de la falta de intercomunicación que se da entre dispositivos ubicuos para poner de relieve la importancia de que los diseñadores de estas plataformas consideren el ecosistema del usuario con todos los dispositivos que utiliza como un todo, y de esta manera lograr una la mejora en la experiencia de uso.

A la charla de Manuel Pérez-Quiñones sucedieron otras a lo largo de la mañana en las que los temas estuvieron relacionados con la búsqueda de información y la minería de datos. La tarde estuvo dedicada al tutorial que impartió Ricardo Baeza-Yates, Vicepresidente de Yahoo! Research, con el título "Introducción a la Minería de Web" y que finalizó la tarde del día siguiente. En total fueron 3 horas de clase con uno de los mayores expertos en minería, un tiempo que se hizo corto para todo lo que se puede aprender sobre este tema. El profesor Baeza-Yates dejó clara la necesidad de aplicar técnicas de minería en los distintos tipos de datos de la Web: aquellos que proceden de los contenidos (textuales y multimedia), los relativos a la estructura de hiperenlaces y en los que se generan con el uso que se hace de la Web (las consultas en los buscadores y la navegación de los usuarios, entre otros). Todos estos datos proporcionan una fuente de información muy potente porque nos permiten conocer cómo es la Web y cómo se utiliza, y esta información puede y debe utilizarse para mejorar la Web y hacerla más útil a las personas. La charla estuvo apoyada por muchos gráficos con distintas técnicas de visualización que sólo en sí mismos ya resultan interesantes, además de aclarativos para entender la estructura de la Web. El tutorial centró buena parte de su atención en la Web 2.0 y en cómo utilizar la información que los usuarios generan para mejorar el diseño de los sitios Web e incluso para predecir la intención de los usuarios en sus consultas y así mejorar los resultados de los buscadores.

En la siguiente jornada asistimos a una buena cantidad de charlas sobre temáticas variadas; la mañana comenzó con varias conferencias sobre TIC para el desarrollo, continuó con el tema de la Web semántica y finalizó con ponencias sobre ingeniería Web y comunidades. Esa tarde tuvimos como ponente invitado a Fabrizio Silvestre, investigador del ISTI en el CNR de Pisa, Italia, que dio una charla sobre cómo el análisis de las búsquedas realizadas en el pasado nos ayudan adoptar mejoras para el futuro usando técnicas de minería de datos. Para demostrarlo nos presentó varios ejemplos de resultados que han obtenido en los últimos años con el uso de técnicas de *caching*, es decir, almacenar en memoria rápida resultados que se usarán frecuentemente. La tarde finalizó con la segunda parte del tutorial de Ricardo Baeza-Yates.

El último día, la conferencia invitada estuvo a cargo de Mariano Consens, profesor en la Universidad de Toronto y especialista en gestión de datos. Su charla fue más hacia la Web Semántica, pues trató sobre la gestión de datos que están vinculados en la Web refiriéndose a cómo distintos conjuntos de datos están unidos entre sí por enlaces en RDF que se identifican por URIs. La red que se forma con estos vínculos puede ser explotada para mejorar las capacidades de los navegadores y de los buscadores y para crear nuevos escenarios, nuevas aplicaciones y nuevos *mashups*. Consens puso énfasis en algunos de los retos que supone esta concepción de la Web: la gestión de los inter-enlaces (*interlinks*), los metadatos, y la vinculación entre conjuntos de datos. Estos retos han sido abordados por LinkedMDB, una base de datos de películas que tiene un gran número de inter-enlaces a otras bases de datos y a otros sitios Web sobre películas.

A la charla de Consens siguió una sesión sobre la investigación sobre la Web en Latinoamérica. Hubo representación de Chile, Brasil y Argentina, y un interesante debate al final de las ponencias. Con esta sesión se puso fin a tres días intensos. Las actas han sido publicadas en Cd-rom por la IEEE Computer Society Press y las presentaciones de los ponentes invitados están disponibles en abierto en el sitio web de LA-Web 2008: `http://www.cwr.cl/la-web2008/`

El presidente de la conferencia y coordinador del comité de seguimiento de LA-Web, Ricardo Baeza-Yates, nos invitó a participar en el próximo, cuya sede será en México. No dudamos que LA-Web 2009 será de nuevo un éxito, aunque Claudine Badue, Elias Oliveira y Alberto Ferreira, comité local en Vila Velha, han puesto el listón bien alto. Además de la calidad de las charlas, la *moqueca capixaba* y las *caipirinhas* nos dejaron un buen sabor de boca de LA-Web 2008. BITS

III Alberto Mendelzon Workshop on the Foundations of Data Management

Arequipa, Perú, 12 al 15 de mayo, 2009
http://db.cs.ualberta.ca/amw09/

Pablo Barceló
DCC, Universidad de Chile

Este workshop es una iniciativa de la comunidad latinoamericana de investigadores identificados con las áreas de bases de datos, manejo de información y la Web, a la cual nuestro amigo Alberto Mendelzon contribuyó con enorme generosidad. Nuestro objetivo es establecer en el cono sur una instancia científica periódica de alto nivel en los aspectos fundamentales del área. Creemos que esta es una excelente forma de mantener viva la memoria de Alberto, y a la vez de de incrementar y solidificar la investigación científica en la región. Como en los dos eventos anteriores, otra de nuestras principales motivaciones es ayudar a que los alumnos latinoamericanos interesados en los fundamentos de bases de datos y la Web (especialmente los alumnos de máster y doctorado) tengan la oportunidad de interactuar con algunos de los más destacados expertos mundiales del área.

Este año hemos decidido realizar el workshop en Arequipa, Perú, desde el 12 al 15 de mayo. El primer workshop (12 al 14 de mayo) consistirá de 3 tutoriales invitados, algunas charlas invitadas, más los artículos aceptados por el comité. Luego habrá un workshop de alumnos graduados (en español, el 15 de mayo). Este también tendrá un par de tutoriales invitados.

Arequipa no es sólo la segunda ciudad más poblada del Perú y un importante centro turístico, sino también el lugar donde se encuentran algunas de las más reconocidas universidades peruanas que realizan docencia e investigación en ciencia de la computación. A través de este workshop esperamos que una buena parte de la gran cantidad de los alumnos peruanos que estudian ciencia de la computación se interesen en los temas relacionados con las bases de datos, manejo de información y la Web.

Entre los atractivos turísticos de Arequipa se puede mencionar su localización, en las contrafuertes de varios volcanes pertenecientes a los Andes Occidentales. Particularmente impresionante es el Cañón del Colca, el más profundo del mundo, a unas horas de Arequipa. A lo largo del valle se pueden ver innumerables construcciones prehispánicas, y en lo alto del cañón es usual observar cóndores. Además, el centro de la ciudad de Arequipa, declarado Herencia Cultural por la UNESCO el año 2000, combina iglesias barrocas con mansiones de la época colonial. Destacan la catedral, un imponente edificio del siglo XVI, y el Monasterio de Santa Catalina. La ciudad es además famosa porque la mayoría de sus casas está construída en piedra volcánica de color blanco, por lo cual a veces recibe el nombre de "Ciudad Blanca". Además Arequipa está a corta distancia de dos de los principales centros turísticos del Perú: Cusco y Machu-Picchu, y Nasca.

ALBERTO MENDELZON

Pero, ¿quién es Alberto Mendelzon y por qué el workshop tiene su nombre? Alberto Mendelzon, sin lugar a dudas, es uno de los más influyentes científicos en el área de las bases de datos. Esto, ciertamente, no es un logro menor considerando la gran cantidad de "superestrellas" que han realizado investigación en el área. Por ejemplo, de los 25 científicos de la computación con mayor índice-h *(h-index)* al menos cinco han realizado investigación prolongada en el área: Héctor García-Molina, Jeff Ullman, Christos Papadimitriou, Alon Halevy y Jennifer Widom. Además, García-Molina y Ullman son los dos científicos de la computación con mayor índice-*h* (89 y 87 ¡respectivamente!). El índice *h* de Alberto es también impresionante: 43 a la fecha, lo que significa que 43 de sus artículos tienen al menos 43 citaciones. Además, cada uno de los cinco artículos más influyentes de Alberto tiene al menos 400 citaciones, mientras que su artículo más citado tiene 640.

Pero claro, todo esto son sólo fríos números, y a nosotros nos gustaría transmitir en unos pocos párrafos, al menos, una parte de lo que fue Alberto como investigador y persona. Primero, los datos biográficos: Alberto nació en Buenos Aires, Argentina, casi 58 años atrás. Recibió su grado de Ph.D. en Princeton en 1979, y tras pasar un año como pos-doc en IBM Watson las emprendió hacia la Universidad de Toronto, lugar que nunca dejaría. Alberto murió en junio de 2005 producto de un cáncer.

Quizá el trabajo más famoso de Alberto es el que realizó durante sus años de Ph.D. (supervisado por el mismísimo Ullman), donde delineó de manera definitiva el posterior desarrollo de la teoría y la práctica de las bases de datos relacionales. El aporte más significativo de este trabajo es haber presentado por primera vez un método –llamado *chase*– para verificar implicación entre restricciones de integridad. Tal método se halla actualmente diseminado a través de toda la literatura en bases de datos, y es utilizado diariamente de forma directa o indirecta por cada persona que diseña bases de datos. Es además el método utilizado por los sistemas comerciales para verificar la consistencia y corrección del diseño de los datos. Pero ciertamente el aporte de Alberto a las bases de datos no se detuvo ahí: sus artículos han delineado e influido fuertemente en muchas otras áreas, incluyendo revisión de conocimiento y actualización de bases de datos, lenguajes de consultas para bases de datos de grafos, consultas en la Web, utilización de vistas en integración de datos, etc. En 2007, ACM PODS, la conferencia más prestigiosa en el área de los fundamentos de las bases de datos, estableció el Alberto Mendelzon *test-of-time-award* en memoria de Alberto.

Por cierto, Alberto no sólo fue un gran científico sino también una persona muy especial para todos aquellos que compartieron con él. Famoso por su modestia y sentido del humor, y admirado por su generosidad, no puede ser mejor descrito que por sus propios colegas y alumnos. Entre ellos me gustaría destacar el testimonio de Moshe Vardi, otra "superestrella": "Alberto no sólo fue tremendamente respetado por sus colegas debido a su trabajo científico, sino también extremadamente querido por sus modos sencillos, sentido del humor y cálida personalidad. Aunque sólo nos veíamos una o dos veces al año, para mí no era sólo un buen colega sino que también un amigo. Lo extrañaré mucho".

Alberto fue particularmente generoso con la comunidad científica latinoamericana dedicada a la computación, de la cual él sin duda se sentía parte. Cada vez que pudo, Alberto visitó países de la región (Argentina, Brasil, Chile, entre otros) para establecer colaboración científica o tan sólo para dictar una charla. Además Alberto ayudó a una larga lista de estudiantes y profesores a llegar a Toronto, ya sea como alumnos de máster, doctorado o posdoctorado. Esta es la razón por la que el workshop recibe su nombre. BITS

PROGRAMA DE DOCTORADO EN CIENCIAS MENCIÓN COMPUTACIÓN

Departamento de Ciencias de la Computación
Facultad de Ciencias Físicas y Matemáticas

Formamos especialistas con amplio dominio de la Ciencia de la Computación capaces de hacer aportes originales y tangibles a la disciplina.

- Hoy 30 estudiantes de Argentina, Bolivia, Colombia, Chile, Francia y Perú entre otras nacionalidades cursan su Doctorado con nosotros.
- Nuestro cuerpo académico consta de 18 doctores más académicos de jornada parcial de alta especialización y conocimiento de la industria.
- Al año recibimos la visita de decenas de investigadores extranjeros y realizamos intercambios regulares de graduados y académicos con diversas universidades como de Lisboa, UPM, Pompeu Fabra, INRIA (con sus diversas sedes), École des Mines de Nantes, Londres, Duisburg, Bolzano, Vienna, Toronto, Columbia.
- Contamos con modernas instalaciones y laboratorios, y cómodas oficinas para nuestros estudiantes.
- Nuestros graduados hoy se desempeñan en universidades nacionales y extranjeras, o realizan estudios en grandes empresas y laboratorios de investigación.

FILOSOFÍA DCC

- Los estudiantes son el centro de nuestro quehacer.
- Estimulamos a nuestros alumnos a que conozcan el desarrollo de la investigación internacional en su área, mediante la inclusión de sus trabajos en conferencias internacionales o con pasantías en centros de excelencia mundial.
- Estudiantes y profesores interactúan e investigan en un ambiente de colaboración, cordialidad y respeto.

LINEAS DE ESPECIALIZACIÓN

Bases de Datos Multimedia, Criptografía Aplicada, Diseño de Algoritmos y Estructuras de Datos, Informática Educativa, Ingeniería de Software, Investigación de la Web, Lenguajes de Programación, Lógica para la Ciencia de la Computación y Aspectos Formales, Modelamiento Geométrico, Recuperación de la Información, Redes, Sistemas Distribuidos y Paralelismo, Seguridad Computacional, Sistemas Colaborativos, Triangulaciones: algoritmos, visualización y aplicaciones interdisciplinarias.

PROGRAMA Y REQUISITOS

Entregamos una formación básica de posgrado y en investigación con cursos especializados. La tesis es el núcleo central: trabajo de investigación que contribuya de forma original y tangible al conocimiento de la disciplina. Duración estimada de la tesis: 2 años; duración total del programa: entre 3,5 a 4 años. Requisito de postulación: poseer el grado de Licenciado en Ciencias mención Computación o algún título o grado equivalente.

MAYOR INFORMACIÓN

estudios@dcc.uchile.cl

Coordinador de Posgrado: Profesor Claudio Gutiérrez.

REVISTA
BITS de Ciencia
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE CHILE

Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl/revista

dcc@dcc.uchile.cl