

Recuperación Eficiente de Datos en Bases de Datos Masivas

Luis Britos, María E. Di Gennaro, Jacqueline Fernández, Veronica Gil-Costa, Fernando Kasián, Jair Lobos, Verónica Ludueña, Marcela Printista, Nora Reyes, Patricia Roggero

LIDIC, Dpto. de Informática, Fac. de Cs. Físico Matemáticas y Naturales, Universidad Nacional de San Luis
{lebritos, mdigena, jmfer, gvcosta, fksian, vlud, mprinti, nreyes, proggero}@unsl.edu.ar, jairlobos@gmail.com

Edgar Chávez

Centro de Investigación Científica y de Educación Superior de Ensenada, México

elchavez@cicese.mx

Claudia Deco

Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario

deco@fceia.unr.edu.ar

Resumen

Los sistemas de información actuales necesitan realizar búsquedas eficientes sobre diferentes tipos de datos, tales como texto libre, audio, video, secuencias de ADN, etc.. Dada una consulta, el objetivo de un sistema de recuperación de información es obtener lo que podría ser útil o relevante para el usuario, en general usando una estructura de almacenamiento especialmente diseñada para responderla de manera eficiente.

Así, nuestra línea de investigación tiene como principal objetivo desarrollar herramientas eficientes para sistemas de información sobre bases de datos masivas, conteniendo datos multimedia. Con este fin, se investigan nuevas técnicas que soporten la interacción con el usuario, nuevas estructuras de datos (índices) capaces de manipular eficientemente datos multimedia y que permitan manejar bases de datos masivas de este tipo de datos.

Palabras Claves: *Recuperación de Información, Computación de Alto Desempeño, Grandes Bases de Datos.*

Contexto

Esta línea de investigación se encuentra enmarcada dentro de la continuación del Proyecto Consolidado 30310 de la Universidad Nacional de San Luis (UNSL) y en el Programa de Incentivos (código 22/F034): “Tecnologías Avanzadas Aplicadas al Procesamiento de Datos Masivos”, dentro de la línea “Recuperación de Datos e Información”, desarrollada en el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la UNSL.

En este contexto, se pretende aportar a la incorporación de información no estructurada en los procesos de toma de decisiones y resolución de problemas, no considerados en los enfoques clásicos.

Dentro de este contexto nuestra línea se dedica, principalmente, al diseño de índices eficientes que

sirvan de apoyo a sistemas de recuperación de información orientados a conjuntos masivos de datos multimedia. Se espera así contribuir a estos sistemas obteniendo índices más eficientes para memorias jerárquicas, dinámicos, con E/S eficiente, escalables (capaces de manejar grandes volúmenes de datos), considerando técnicas de computación de alto desempeño (HPC).

1. Introducción y Motivación

Con el uso masivo de internet, estamos en presencia de un fenómeno donde la rápida aceleración tanto del crecimiento del volumen de datos capturados y almacenados, como la creciente variación en los tipos de datos requeridos, hace que las técnicas tradicionales para el procesamiento, análisis y obtención de información útil deban ser redefinidas para formular nuevas metodologías de abordaje.

Los sistemas de computación tradicionales hacen uso intensivo de información estructurada, es decir datos elementales o estructuras, generadas con un formato específico. Una característica principal en estos casos, es que la estructura o formato de esta información puede ser fácilmente interpretada y directamente utilizada por un programa de computadora. Pero el hecho de restringirse al uso de este tipo de información conduce, muchas veces, a representar una visión parcial del problema y dejar fuera información que podría ser importante para la resolución efectiva del mismo. En este contexto gran parte de la información que se requiere para la toma de decisiones y la resolución de problemas de índole general proviene de información no estructurada.

En general, para responder eficientemente consul-

tas para recuperación de información sobre bases de datos multimedia se utilizan diferentes métodos de acceso o índices [6], principalmente por el gran volumen de datos con el que se trabaja. Algunos poseen características que los hacen indicados para aplicaciones reales: eficientes, dinámicos, escalables y resistentes a la *maldición de la dimensión*.

Un enfoque prometedor para sistemas de recuperación usando búsqueda por similitud es una búsqueda basada en contenidos, la cual usa el dato multimedia mismo. Para calcular la similitud entre dos objetos multimedia, se debe definir una función de distancia. Dicha función mide la similitud, o más bien la disimilitud, entre dos objetos. En muchos casos para modelar la similitud de objetos multimedia se transforman los objetos en puntos multidimensionales de un espacio vectorial, el cual es un tipo particular de espacio métrico. Cada objeto es representado por un vector de características o descriptor, generalmente de alta dimensionalidad. Sobre espacios vectoriales existen numerosas funciones de distancia. El tipo de aplicación, las características a explotar o la dimensionalidad son aspectos a considerar para definir la mejor función de distancia a utilizar.

El concepto de búsqueda por similitud se puede definir a partir del de espacios métricos, que da un marco formal independiente del dominio de aplicación. Un espacio métrico está compuesto por un universo \mathcal{U} de objetos y una función de distancia $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^+$, que satisface las propiedades que la hacen una métrica. Las consultas por similitud, sobre una *base de datos* $\mathcal{S} \subseteq \mathcal{U}$, son usualmente de dos tipos: *Búsqueda por rango* y *Búsqueda de los k vecinos más cercanos*. La función de similitud (distancia) mide el mínimo esfuerzo (costo) necesario para transformar un objeto en otro. Dependiendo de los tipos de datos multimedia reales la función de similitud puede ser muy compleja. En particular, para poder ahorrar cálculos de distancia es importante que la distancia satisfaga la desigualdad triangular.

Si la base de datos \mathcal{S} posee n objetos, las consultas pueden ser respondidas llevando a cabo n evaluaciones de distancia. Sin embargo, en la mayoría de las aplicaciones las distancias son costosas de computar (por ej.: comparación de huellas digitales), por lo que la búsqueda secuencial no sirve para conjuntos masivos de datos, que son los tamaños que se están tornando más habituales en estos repositorios de datos multimedia. Así debemos preprocesar la base de datos, construyendo un índice, para responder a las consultas con la menor cantidad de

cálculos de distancia. Además, es probable que la base de datos, el índice o ambos no puedan almacenarse en memoria principal con lo cual se debe considerar minimizar también el número de operaciones de E/S, tener siempre presente la jerarquía de memorias y tratar de lograr mayor eficiencia a través de técnicas paralelas.

Entonces, el foco de esta propuesta es mejorar herramientas de recuperación, desarrollando nuevas técnicas que soporten la interacción con el usuario, diseñando estructuras de datos (índices), capaces de manipular eficientemente grandes volúmenes de datos y facilitando la realización de diferentes operaciones, de modo de acercarse al nivel de desarrollo que se posee en bases de datos tradicionales.

2. Líneas de Investigación

Se pretende investigar sobre distintos aspectos de los sistemas de recuperación de información multimedia: diseñar nuevos índices, definir representaciones que reflejen características de interés de los objetos y manejar distintas operaciones sobre estos tipos de bases de datos, considerando trabajar eficientemente sobre grandes volúmenes de datos.

Diseño de Índices

Un catálogo importante de índices para espacios métricos aparece en [6]. La mayoría usan la desigualdad triangular para evitar el análisis secuencial de la base de datos. La distancia entre la consulta q y los objetos de la base de datos puede ser estimada calculando de antemano algunas distancias a objetos distinguidos llamados *pivotes* y sin calcular las distancias reales desde q a los objetos de la base de datos durante una búsqueda. Otra técnica común es indexar a través de una partición del espacio en regiones denominadas *particiones compactas*.

Existen dos posibles situaciones por el tipo de base de datos con la que se va a trabajar, que determinan una característica importante que debe tener el índice que la manipulará: los objetos de la base de datos se conocen de antemano y por lo tanto el índice se creará de una sola vez y se realizarán consultas sobre él (índices estáticos); o no se conocen los objetos de antemano y por lo tanto el índice se debe ir creando a medida que arriban los elementos y preferentemente de manera incremental (índices dinámicos). Las estructuras estáticas se benefician desde el conocimiento de la base de datos seleccionando los mejores puntos de referencia para una estructura de datos determinada, lo cual no es posible en las estructuras de datos dinámicas donde tanto los objetos

como las consultas arriban al azar.

Cuando se trabaja sobre conjuntos masivos de datos una manera de lograr eficiencia en las operaciones sobre los índices es aplicando técnicas de computación de alto desempeño y en otros casos mediante la adaptación o diseño de las estructuras para que sean concientes de la jerarquía de memorias, minimizando no sólo la cantidad de cálculos de la función de distancia, sino también el número de operaciones de E/S. Además, el estudio detallado de qué hace que un índice sea intrínsecamente más eficiente ayuda a mejorar los costos de las operaciones.

El interés está en mejorar el desempeño de índices dinámicos jerárquicos (árboles), que es el caso de algunos de los índices para espacios métricos. Estos índices dinámicos, en general, se construyen incrementalmente vía inserciones. De tal manera, la raíz del árbol es el primer objeto que llega, y esto se repite recursivamente en cada nivel del árbol.

Como ya mencionamos, el DiSAT es un índice para espacios métricos muy eficiente, considerando el número de cálculos de distancias realizados tanto en construcción como en búsquedas. La desventaja del DiSAT es que no admite inserciones ni eliminaciones y no es posible construirlo incrementalmente. Sin embargo, una opción para transformarlo en una estructura dinámica es mediante la aplicación de la técnica de Bentley y Saxe [2] que permite lograr dinamismo a partir de una estructura estática cuando la búsqueda sobre ella cumple con ser un problema que se puede descomponer en partes independientes (descomponible). Por lo tanto, se está desarrollando una versión dinámica del DiSAT para resolver eficientemente el problema, no “descomponible”, de la búsqueda de k -vecinos más cercanos.

En muchos casos los volúmenes de información con los que se debe trabajar (millones de imágenes en la Web), hacen necesario que los índices sean almacenados en memoria secundaria. En este caso, para hacerlos eficientes, no sólo se debe considerar que durante las búsquedas se realice el menor número de cálculos de distancia sino también, dado el costo de las operaciones sobre disco, se efectúe la menor cantidad posible de operaciones de E/S. Por ello, en esta línea nos hemos dedicado a diseñar índices especialmente adaptados para trabajar en memoria secundaria, logrando un buen desempeño de los mismos, principalmente en las búsquedas.

Hemos diseñado e implementado las siguientes estructuras *DSACL*-tree* y el *DSACL+-tree* [3], las cuales son optimizaciones para memoria secundaria

de la estructura propuesta en [1] y demostraron ser competitivas frente a otras de las estructuras conocidas tales como el *M-tree* y *DSA*-tree* y *DSA+-tree* [11]. Además, existen nuevas propuestas en evaluación que prometen ser aún más adecuadas para memoria secundaria. Por otro lado, nos proponemos optimizarlas todavía más gracias a la aplicación de técnicas de computación de alto desempeño, aplicando y comparando distintas estrategias de paralelización con el fin de determinar la más adecuada.

Procesamiento de Consultas en Redes P2P

Los motores de búsqueda Web son sistemas ideados para hacer frente a una gran tasa de consultas dinámicas y exigentes, posiblemente del orden de muchos cientos de miles de consultas por segundo. Para soportar redundancia y tolerancia a fallos, grandes motores de búsqueda funcionan en múltiples centros de datos distribuidos geográficamente. La intensidad del tráfico de consultas se caracteriza por el comportamiento impredecible de los usuarios que son por lo general muy reactivos a los eventos de todo el mundo. Por otro lado, muchos usuarios acceden a Internet desde su hogar a través de “set-top-boxes” (STB) conectados a una red ISP (Internet Service Provider). Estas cajas son equipos reales que la gente suele mantener encendida sin realizar cálculos exigentes y de ese modo se desperdicia energía. La infraestructura basada en STB ha sido llamado “nanodatacenter”. En estado de reposo estos sistemas consumen aproximadamente el 50-80 % de la potencia consumida bajo carga máxima [17].

Por otra parte, los nanodatacenters evitan los costos de enfriamiento y reducen el consumo de energía de la red. Varios trabajos recientes [15, 16] han propuesto tomar ventaja de esta potencia disponible de cálculo mediante la organización de STBs como redes Peer-to-Peer (P2P), que se utilizan para apoyar los servicios de intercambio de grandes volúmenes de datos entre los usuarios.

Por lo tanto, el objetivo en este caso es analizar la colaboración de los proveedores de servicios Internet (ISP), los motores de búsqueda Web y los usuarios finales en el hogar [14, 7] y evaluar la posibilidad de un modelo global que permita disminuir el consumo de energía de los centros de datos. Los usuarios finales en el hogar cooperan compartiendo recursos energéticos para apoyar el procesamiento global de consultas por ser parte de una memoria caché de resultados de la consulta distribuida. Para ello, cada “peer” mantiene una caché de resultados de consultas. La caché tiene una sección *local*

(guarda consultas realizadas por el peer) y una sección *global* (guarda consultas realizadas por otros peers). La idea básica es que las consultas almacenadas en caché contienen los top- K resultados obtenidos desde el motor de búsqueda Web. Cuando llega una nueva consulta desde un peer (STB), ésta se busca en la caché local del mismo. Si no se encuentran resultados se envía la consulta al peer responsable. Si éste tiene resultados se los retorna al peer solicitante, sino continúa la búsqueda en el siguiente peer responsable hasta llegar al motor de búsqueda Web.

Consultas sobre Bases de Datos Multimedia

Aunque las operaciones más comunes sobre bases de datos multimedia son las búsquedas por rango o de k -vecinos más cercanos, existen otras operaciones de interés como las distintas variantes del *join* por similitud. Para estas operaciones se consideran dos bases de datos A y B , ambas subconjuntos del mismo universo del espacio métrico \mathcal{U} . El resultado de cualquier operación de *join* por similitud entre A y B obtiene el conjunto de pares formados por un objeto de A y otro de B , tales que entre ellos se satisface el predicado de similitud considerado. Las variantes más conocidas del *join* por similitud son: el *join* por rango, el *join* de k -vecinos más cercanos y el *join* de vecino más cercano; entre otras.

Formalmente, dadas $A, B \subseteq \mathcal{U}$, se define el *join por similitud* entre A y B ($A \bowtie_{\Phi} B$) como el conjunto de todos los pares (x, y) , donde $x \in A$ e $y \in B$; es decir, $(x, y) \in A \times B$, tal que entre x e y se satisface el criterio de similitud considerado Φ . De acuerdo al criterio de similitud el *join* puede llamarse: *Join por rango* o *Join de k -vecinos más cercanos*. Existen dos situaciones distintas sobre las que se puede trabajar, para resolver el *join* por similitud: que ambas bases de datos se encuentren indexadas, por separado; o que ambas bases de datos se indexen conjuntamente, con un índice diseñado para el *join*. Como calcular cualquiera de las variantes del *join* por similitud de manera exacta es muy costoso [13], vale la pena analizar posibilidades de obtener más rápidamente una respuesta aproximada al *join*, logrando una respuesta rápida y de buena calidad.

PostgreSQL es el primer sistema de base de datos que permite realizar consultas por similitud sobre algunos atributos, particularmente indexación para búsquedas de k -vecinos más cercanos (KNN-GiST indexes). Estos índices pueden ser usados sobre texto, comparación de ubicación geoespacial, etc.. Sin embargo, los índices K-NN GiST proveen plantillas para índices con estructura de árbol balanceado (B-

tree, R-tree), aunque el balance no siempre es bueno para los índices que se utilizan en búsquedas por similitud [4]. Además, no se dispone de este tipo de consultas para todo tipo de datos métricos. Así, es importante proveer un DBMS para bases de datos métricas que maneje todos los posibles datos métricos y las operaciones de interés sobre ellos [8].

3. Resultados

Se ha comprobado experimentalmente que el DSAT es mejor que su versión estática [11], por dejar como “vecinos” a objetos alejados, permitiendo así avanzar en la exploración espacial a “pasos más grandes”, obteniendo así el DiSAT [5]. Además, se obtuvo un nuevo método de eliminación para el DSAT [9].

Se implementaron dos versiones: DSACL*-tree y DSACL+-tree, que trabajan con grandes volúmenes de datos, por haber sido diseñadas para memoria secundaria y que mostraron ser competitivas contra otras estructuras diseñadas para tal fin [3]. Se espera lograr una implementación paralela eficiente de estos índices. Se ha logrado mayor eficiencia en otros índices [10, 12] gracias al uso de técnicas paralelas.

Se han propuesto estrategias de “caching” P2P de dos niveles para responder consultas en la Web, a través de una adecuada plataforma de servicios completamente distribuida que utiliza los componentes ya existentes en las redes. Esta solución reduce significativamente el tráfico del proveedor de servicios para obtener resultados de consultas en el motor de búsqueda Web [14, 7].

4. Formación de Recursos

Para contribuir al desarrollo de sistemas de recuperación de información multimedia, se están capacitando los siguientes investigadores:

Tesis de Doctorado en Cs. de la Computación: un integrante está desarrollando su tesis sobre “Planeación de Capacidad en Centros de Datos para Sistemas Escalables para la Web” (con beca de posgrado-UNSL), otro se encuentra definiendo su plan de doctorado sobre diseño y optimización de índices, para aplicaciones de minería de datos multimedia.

Tesis de Maestría en Cs. de la Computación: una sobre índices dinámicos para búsqueda eficiente en memoria principal, dos sobre índices dinámicos eficientes para datos masivos: una que aplica técnicas de computación de alto desempeño y la otra sólo aplicando optimizaciones considerando la jerarquía

de memorias, y una sobre un DBMS para bases de datos métricas.

Referencias

- [1] M. Barroso, N. Reyes, and R. Paredes. Enlarging nodes to improve dynamic spatial approximation trees. In *Proc. of the 3rd SISAP*, pages 41–48. ACM Press, 2010.
- [2] J. Bentley and J. Saxe. Decomposable searching problems: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.
- [3] L. Britos, A. M. Printista, and N. Reyes. Dynamic spatial approximation trees with clusters for secondary memory. In G. Simari, P. Pesado, and J. Paganini, editors, *XVI CACIC Selected Papers*, Computer Science & Technology Series, pages 205–215. Editorial UNLP, 2011.
- [4] E. Chávez, V. Ludueña, and N. Reyes. Revisiting the VP-forest: Unbalance to improve the performance. In *Proc. de las JCC08*, page 26, 2008.
- [5] E. Chávez, V. Ludueña, N. Reyes, and P. Roggero. Faster proximity searching with the distal sat. Technical report, Universidad Nacional de San Luis, 2013.
- [6] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM*, 33(3):273–321, September 2001.
- [7] N. Hidalgo, E. Rosas, V. Gil-Costa, and M. Marin. Assessing energy efficiency in isp and web search engine collaboration. In *The Tenth International Symposium on Frontiers of Information Systems and Network Applications (FINA 2014)*, Victoria, Canada, May 2014. Por aparecer.
- [8] F. Kasián and N. Reyes. Búsquedas por similitud en PostgreSQL. In *Actas del XVIII Congreso Argentino de Ciencias de la Computación (CACiC)*, pages 1098–1107, Bahía Blanca, Argentina, October 2012. Universidad Nacional del Sur.
- [9] F. Kasián, V. Ludueña, N. Reyes, and P. Roggero. An efficient alternative for deletions in dynamic spatial approximation trees. *Journal of Computer Science & Technology*, 14(1):39–45, April 2014.
- [10] V. Mancini, F. Bustos, V. Gil-Costa, and A. M. Printista. Data partitioning evaluation for multimedia systems in hybrid environments. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*, pages 321–326, 2012.
- [11] G. Navarro and N. Reyes. Dynamic spatial approximation trees. *Journal of Experimental Algorithmics*, 12:1–68, 2008.
- [12] C. Ochoa, V. Gil-Costa, and A. M. Printista. Suffix array performance analysis for multi-core platforms. In *4th International SuperComputing Conference in Mexico (ISUM)*, 2013. En CD.
- [13] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *JDA*, 7:18–35, March 2009. doi:10.1016/j.jda.2008.09.012.
- [14] E. Rosas, N. Hidalgo, M. Marin, and V. Gil-Costa. Web search results caching service for structured {P2P} networks. *Future Generation Computer Systems*, 30(0):254–264, 2014. Special Issue on Extreme Scale Parallel Architectures and Systems, Cryptography in Cloud Computing and Recent Advances in Parallel and Distributed Systems, {ICPADS} 2012 Selected Papers.
- [15] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens. Aren: A popularity aware replication scheme for cloud storage. In *ICPADS*, pages 189–196. IEEE Computer Society, 2012.
- [16] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens. Caju: A content distribution system for edge networks. In I. Caragiannis, M. Alexander, R. M. Badia, M. Cannataro, Al. Costan, M. Danelutto, F. Desprez, B. Krammer, J. Sahuquillo, S. L. Scott, and J. Weidendorfer, editors, *Euro-Par Workshops*, volume 7640 of *Lecture Notes in Computer Science*, pages 13–23. Springer, 2012.
- [17] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez. Greening the internet with nano data centers. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 37–48, New York, NY, USA, 2009. ACM.